S. ROBERTS

The Fourth Book of OHIO SCIENTIFIC

VIP Very Important Programs for OHIO SCIENTIFIC

Acknowledgements

ELCOMP Microcomputer Magazine Vol. 1 Nr. 1 Vol. 3 Nr.
PERSONAL COMPUTER WORLD MAGAZINE, Great Britain

All the programs, compiled in this book can be used without modification on your Ohio C1P or C4P (or C1PMF/C4PMF) Computer. Basicly it is Microsoft BASIC, so with minor changes you can run these programs on any BASIC-Computer.

For anyone interested, a list of dealers may be obtained by writing to

Ohio Scientific, Inc.
1333 South Chillicothe Rd.
Aurora, OH 44 202

We also wish to thank John and Mary Neally and Franz Ende for their help in completing this book.

ISBN 3–921682–78–9

This book is published as a service to all Ohio Scientific users. No liability is assumed with respect to the use of the information herein.

HOFACKER VERLAG, Ing. W. Hofacker GmbH, Postbox 75 04 37, D–8000 München 75, West Germany

US-Distribution:

ELCOMP Publishing, Inc., 3873L Schaefer Avenue, Chino, CA 91710

S. ROBERTS

The Fourth Book of OHIO SCIENTIFIC

VIP Very Important Programs

for OHIO SCIENTIFIC

TABLE of CONTENTS

USEFUL PROGRAMS

RAM Test

RAM TEST - The program POKES and PEEKS random numbers. If no defects are found, the screen will show "(X.)LOOP(Y)DEFECTS". If a defect is found, the screen will show the number of defects and the number of the IC.

```
 10 REM RAM-TEST
 20 FORI=1TO30:PRINT:NEXT
 30 DIMFL(32),FH(32)
 40 PRINT"   RAM-MEMORY TEST":PRINT
 50 FORI=1TO6:PRINT:NEXT
 60 INPUT"START ADDRESS";A:PRINT
 70 INPUT"END ADDRESS";E:PRINT
 80 IFA<2000THEN60
 90 IFA>ETHEN60
 100 IFE<ATHEN70
 110 IFE>32766THEN70
 120 INPUT"ENTER A RANDOM NUMBER";ZU:PRINT
 130 IFZU=0THEN120
 140 V2=RND(ZU):V1=V2
 150 FORI=ATOE
 160 V1=LOG(V1*V1):V=ABS(INT(V1*100))
 170 IFV>255THEN160
 180 POKEI,V
 190 NEXTI
 200 V1=V2
 210 FORI=ATOE
 220 V1=LOG(V1*V1):V=ABS(INT(V1*100))
 230 IFV>255THEN220
 240 X=PEEK(I)
 250 IFX=VTHEN290
 260 PRINT"ADD";I;"   DIFF";V-X:Z=Z+1
 270 IFABS(V-X)>15THENFH(INT(I/1024))=FH(IN
T(I/1024))+1
```

```
 280 IFABS(V-X)<16THENFL(INT(I/1024))=FL(IN
T(I/1024))+1
 290 NEXTI
 300 D=D+1:PRINTD;". LOOP";Z;"DEFECTS"
 310 FORJ=2TO32
 320 BO=600:IFJ>8THENBO=610
 330 BL=31:BH=15
 340 IFJ>8THENBL=11:BH=19
 350 IFJ>16THENBL=19:BH=27
 360 IFJ>24THENBL=27:BH=35
 370 IFFL(J)=0THEN390
 380 PRINTFL(J);"DEFECT IN IC";J+BL;BO
 390 IFFH(J)=0THEN410
 400 PRINTFH(J);"DEFECT IN IC";J+BH;BO
 410 NEXTJ
 420 PRINT
 430 GOTO140

OK
```

Memory Test

MEMORY TEST - This is a memory test for the Superboard with either 4K or 8K. The program POKES and PEEKS each memory location. If there is a difference encountered, the two numbers will appear at the specific memory location.

```
 10 Y=8191
 20 INPUT"4 OR 8K ";K:IFK=4THENY=4094
 30 P=1030
 40 Q=255
 50 FORX=PTOY
 60 POKEX,Q
 70 NEXTX
 80 FORX=PTOY
 90 Z=PEEK(X)
 100 IFZ<>QTHENGOSUB140
 110 NEXTX:PRINT" PASS USING ";Q
 120 IFQ=0THEN40
 130 Q=0:GOTO50
 140 PRINT" LOCATION "X;" WAS ";Z " NOT ";Q
:RETURN
OK
```

Hex Dump in BASIC

HEX DUMP IN BASIC - This BASIC program displays memory content from a specific area in hexidecimal.

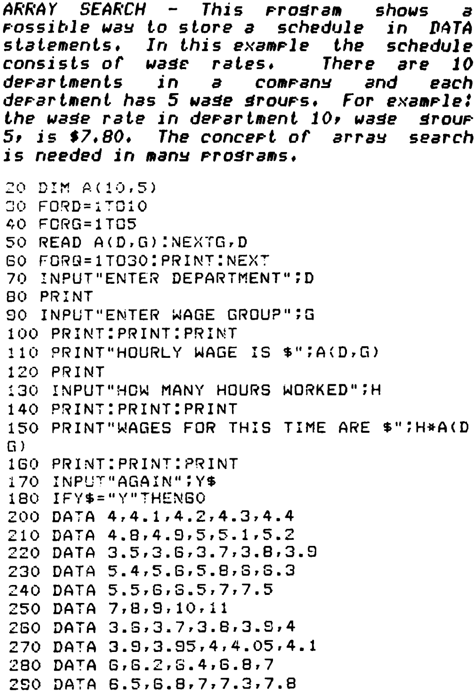
```
 10 REM MEMORY DUMP IN BASIC
 20 FOR X=1 TO 30:PRINT:NEXT
 30 DIM Z$(16)
 40 Z$="0123456789ABCDEF"
 50 PRINT"OUTPUT FROM MEMORY CONTENTS IN HE
X"
 60 PRINT
 70 PRINT"ADDRESS FROM (HEX)"
 80 INPUT A$
 90 PRINT" TO (HEX)"
 100 INPUT B$
 110 PRINT
 120 A=0
 130 B=0
 140 V$=MID$(A$,1,2)
 150 GOSUB430
 160 A=V*256
 170 V$=MID$(A$,3,2)
 180 GOSUB430
 190 A=A+V
 200 V$=MID$(B$,1,2)
 210 GOSUB430
 220 B=V*256
 230 V$=MID$(B$,3,2)
 240 GOSUB430
 250 B=B+V
 260 J=A
 270 V=INT(J/256)
 280 GOSUB490
 290 PRINTV$;
 300 V=J-V*256
 310 GOSUB490
 320 PRINTV$;"  ";
 330 V=PEEK(J)
 340 GOSUB490
 350 PRINTV$;"  ";
 360 J=J+1
 370 IFJ/4<>INT(J/4)THEN330
 380 PRINT"  ";
 390 IFJ/16<>INT(J/16)THEN330
 400 IF J>B THEN STOP
 410 PRINT
 420 GOTO270
 430 FORI=0TO15
 440 IFMID$(V$,1,1)=MID$(Z$,I+1,1)THENV1=I
 450 IFMID$(V$,2,1)=MID$(Z$,I+1,1)THENV2=I
 460 NEXT I
 470 V=V1*16+V2
 480 RETURN
 490 V1=INT(V/16)
 500 V2=V-V1*16
 510 V$=MID$(Z$,V1+1,1)+MID$(Z$,V2+1,1)
 520 RETURN

OK
```

Joystick for CIP

JOYSTICK PROGRAM FOR C1P - This program is written for the Superboard with two connected joysticks. You can move the tanks in eight different directions. Push down on the button to fire at your opponent.

```
 10 FORX=1TO30:PRINT:NEXT
 20 FORX=1TO8:READM(X),T(X):NEXT
 30 A=54040:B=54010:T1=248:T2=248
 40 POKE530,1
 50 POKE57088,127:G=A:T=T1:GM=M1:Y=0:GOSUB1
00:T1=T
 60 POKEA,32:A=A+M:POKEA,T1:IFM<>0THENM1=M
 70 POKE57088,191:G=B:T=T2:GM=M2:Y=1:GOSUB1
00:T2=T
 80 POKEB,32:B=M+B:POKEB,T2:IFM<>0THENM2=M
 90 GOTO50
 100 P=PEEK(57088)
 110 IFP=127ANDY=1THENZ=A:GOSUB230
 120 IFP=127ANDY=0THENZ=B:GOSUB230
 130 M=0
 140 IFP=191THENM=M(1):T=T(1)
 150 IFP=223THENM=M(2):T=T(2)
 160 IFP=239THENM=M(3):T=T(3)
 170 IFP=247THENM=M(4):T=T(4)
 180 IFP=159THENM=M(5):T=T(5)
 190 IFP=207THENM=M(6):T=T(6)
 200 IFP=231THENM=M(7):T=T(7)
 210 IFP=183THENM=M(8):T=T(8)
 220 RETURN
 230 FORX=1TO10:POKEG+X*GM,171
 240 IFPEEK(Z)=171THEN280
 250 POKEG+X*GM,32
 260 NEXT
 270 RETURN
 280 E=42
 290 FORX=1TO5:POKEZ+X,E:POKEZ-X,E:POKEZ-32
,E:POKEZ+32,E
 300 POKEZ-32*X,E:POKEZ+32*X,E:POKEZ-32*X,E
:NEXTX
 310 IFE=42THENE=32:GOTO290
 320 FORX=1TO25:PRINT:NEXT
 330 A$="B"
 340 IFY=0THENA$="A"
 350 PRINT" S C O R E"
 360 PRINT
 370 PRINT" FOR ";A$
 380 PRINT:PRINT:PRINT:PRINT:PRINT
 390 INPUT"ONCE AGAIN (Y/N)";Q$
 400 IFQ$="N"THENEND
 410 RESTORE:GOTO10
 420 DATA -32,248,1,250,32,252,-1,254,-31,2
49,33,251,31,253,-33,255
OK
```

Array Search



ARRAY SEARCH - This program shows a possible way to store a schedule in DATA statements. In this example the schedule consists of wage rates. There are 10 departments in a company and each department has 5 wage groups. For example: the wage rate in department 10, wage group 5, is $7.80. The concept of array search is needed in many programs.

```
20 DIM A(10,5)
30 FORD=1TO10
40 FORG=1TO5
50 READ A(D,G):NEXTG,D
60 FORQ=1TO30:PRINT:NEXT
70 INPUT"ENTER DEPARTMENT";D
80 PRINT
90 INPUT"ENTER WAGE GROUP";G
100 PRINT:PRINT:PRINT
110 PRINT"HOURLY WAGE IS $";A(D,G)
120 PRINT
130 INPUT"HOW MANY HOURS WORKED";H
140 PRINT:PRINT:PRINT
150 PRINT"WAGES FOR THIS TIME ARE $";H*A(D
G)
160 PRINT:PRINT:PRINT
170 INPUT"AGAIN";Y$
180 IFY$="Y"THEN60
200 DATA 4,4.1,4.2,4.3,4.4
210 DATA 4.8,4.9,5,5.1,5.2
220 DATA 3.5,3.6,3.7,3.8,3.9
230 DATA 5.4,5.6,5.8,6,6.3
240 DATA 5.5,6,6.5,7,7.5
250 DATA 7,8,9,10,11
260 DATA 3.6,3.7,3.8,3.9,4
270 DATA 3.9,3.95,4,4.05,4.1
280 DATA 6,6.2,6.4,6.8,7
290 DATA 6.5,6.8,7,7.3,7.8
```

Memory Map

MEMORY MAP -

0000	0	JUMP in Warm-start to BASIC (4C 74 A2) JUMP in cold-start to BASIC (4C 11 BD)
0003	3	Message printer (ABC3)
0008-0009	8-9	USR Program address
000B-000C	11-12	Address of USR Routine
000D	13	Number of nulls after CR
000E	14	Number of signs after last CR
000F	15	Terminal width for auto CRLF
0010	16	Terminal width, divided by commas
0013-005A	19-90	Input buffer
0017	23	Terminal width
0018	24	No. of characters in fields (14)
005F	95	String variable being processed flag (?)
0061	97	?
0064	100	CTRL O Flag (Hi bit= 1) suppresses the printout
0065	101	Sometimes contains $68 (??)
0078	120	Lo byte addresses beginning of BASIC workspace

0079	121	Hi byte addresses beginning of BASIC workspace
0079-007A	121-122	Pointer to the first null of BASIC work memory
007B-007C	123-124	Pointer to the beginning of variable memory
007D-007E	125-126	Pointer to the beginning of the BASIC Array memory
007F-0080	127-128	Pointer to the end of Array, start of free memory area
0080-008A	128-139	Sometimes next line number (?)
0081-0082	129-130	Pointer to (top of free memory) to end of string area
0084	132	LO byte address at end of BASIC workspace
0085	133	Hi byte address at end of BASIC workspace
0085-0086	133-134	Pointer to upper limit of memory, usable by BASIC
0087-0088	135-136	Actual BASIC line number
008F-0090	143-144	DATA pointer
0095-0096	149-150	This is where ADOB leaves address of the variable it found
0097-0098	151-152	Address of variable to be assigned value by OUTVAR(AFC1)
00A1	161	Genl purp JMP instr; put target addr in A2, A3

00AA-00AB	170-171	Points to the pointer of the next BASIC line after LIST
00AD-00AE	173-174	The contents of this pair is printed in decimal by B962
00AE-00AF	174-175	This is where INVAR (AE05) leaves its argument
00BC	188	Get next char in BASIC line
00C2	194	Get current char in B line
00D1-00D7	209-215	Clobbered by OSI Extended Monitor disassembler; kills BASIC
00DE	222	Real Time Monitor(1-enables 0-disables)
00DF	223	Start Countdown timer (1-start 0-stop)
00E0	224	Hours to Countdown
00E0-00E6	224-230	At the moment RAM area not used in the zero page
00E1	225	Minutes to Countddown
00E2	226	Seconds to Countdown
00E6	230	Mask Register for Port 1A DATA-correponding Register
00E7	231	Mask Register for Port 1A CONTROL
00E8	232	Mask Register for Port 1B DATA
00E8-00FF	232-255	At the moment RAM area of BASIC is not used
00E9	233	Mask Register for Port 1B CONTROL
00EA	234	Mask Register for Port 2A DATA

00EB	235	Mask Register for Port 2A CONTROL
00EC	236	Mask Register for Port 2B DATA
00ED	237	Mask Register for Port 2B CONTROL
00EE	238	Mask Register for Port 3A DATA
00EF	239	Mask Register for Port 3A CONTROL
00F0	240	Mask Register for Port 3B DATA
00F1	241	Mask Register for Port 3B CONTROL
00F9	249	Contains largest value (56832) that can be PEEKed
00FB	251	Cassette/Keyboard-Flag for monitor
00FC	252	Plus memory data for the monitor
00FE-00FF	254-255	Address of the actual ROM monitor location
0100-0141	256-321	Stack
0130	304	NMI Routine
01C0	448	IRQ Routine
0200	512	Cursor Position
0201	513	Save character to be printed
0202	514	Temp storage used by CRT driver
0203	515	LOAD-Flag
0205	217	SAVE-Flag
0206	518	Baud Rate for CRT, 00=fast, FF=slow
0207-020E	519-526	Variable execution block-code for screen scroll - not reusable
0212	530	Control C-Flag(If Control C-key is pushed Flag is set)

0213-0216	531-534	Polled Keyboard plus memory and counter
0218	536	Input Vector FFBA
021A	538	Output Vector FF69
021C	540	Control-C Check-Vector FF9B
021E	542	LOAD Vector FFBB
0222-02FB	546-760	Free for small user routines
0224	548	Hi byte address for C driver
0225	549	Lo byte address for C driver
02E5	741	"LIST" (76-enables 10-disables)
02EE	750	"NEW" (78-enables 10-disables)
0705	1797	Control line # list of BASIC (32-enables 44-disables)
0819	2073	"CONTROL C" (173-enables 96-disables)
0898	2200	ROM direction of mask 0 to load at 0220
0B48	2888	Null input jumps out of program(0-disables 27-enables)
0B4D	2893	(28-"REDO FROM START" enable)
0B4E	2894	(11-"REDO FROM START" enable)
0B48-2212	2888-8722	Both 0 & null input to "INPUT" yields empty array or a 0. If Both are 27, then input statement function is normal.
0B9C	2972	58 (comma) is a string input termination. 13-disables

0BA0	2976	44 (colon) is a string input termination. 13-disables
2204	8708	Output flag for peripheral devices (44)
22C6	8902	Determines which register (less 1) RTMON scans
22CD	8909	Hi byte address of PIA for RTMON scanning
22CE	8910	Lo byte address of PIA for RTMON scanning
22D5	8917	USR(X) operation code
22F0	8944	Output flag
22FA	8954	Location of JSR to disk USR(X) routine
22FB	8955	Lo byte address of USR(X) pointer
22FC	8956	Hi byte address of USR(X) pointer
2300	8960	Memory (RAM) page count minus 1
2321	8993	I/O distributor input flag
2322	8994	I/O distributor output flag
2323	8995	Index to current ACIA on 550 board
2324	8996	Location of random seed for RND function
2326	8998	Lo byte address of pointer to disk buffer 1
2327	8999	Hi byte address of pointer to disk buffer 1

2328	9000	Lo byte address of the end (plus 1) of disk buffer area
2329	9001	Hi byte address of the end (plus 1) of disk buffer area
232E-2335	9006-9013	Memory buffered disk I/O bit 6 device parameters
238A	9098	Lo byte address for memory input
238B	9099	Hi byte address for memory input
2391	9105	Lo byte address for memory output
2392	9106	Hi byte address for memory output
23AC	9132	Lo byte address for memory buffered disk input
23AD	9133	Hi byte address for memory buffered disk input
23C3	9155	Lo byte address for memory buffered disk output
23C4	9156	hi byte address for memory buffered disk output
23FD	9213	Lo byte address for memory buffered disk input
23FE	9214	Hi byte address for memory buffered disk input
2416	9238	Lo byte address for memory buffered disk output
2417	9239	Hi byte address for memory buffered disk output
2498	9368	Hi byte address for

		indirect file input (low-00)
2552	9554	Hi byte address for indirect file output (low-00)
24B0	9392	Active State Register for Port 1A Data Register
24B1	9393	Active State Register for Port 1A Control Register
24B2	9394	Active State Register for Port 1B Data Register
24B3	9395	Active State Register for Port 1B Control Register
24B4	9396	Active State Register for Port 2A Data Register
24B5	9397	Active State Register for Port 2A Control Register
24B6	9398	Active State Register for Port 2B Data Register
24B7	9399	Active State Register for Port 2B Control Register
24B8	9400	Active State Register for Port 3A Data Register
24B9	9401	Active State Register for Port 3A Control Register
24BA	9402	Active State Register for Port 3B Data Register
24BB	9403	Active State Register for Port 3B Control Register
25C2	9666	Move left margin to

		the right
25C3	9667	Moves scroll up
25D0	9680	This location contains the cursor character designation
265E	9822	Sector # for USR(X) disk operation
265F	9823	Page count for USR(X) disk write
2660	9824	Lo byte address of memory block for USR(X) disk operation
2661	9825	Hi byte address of memory block for USR(X) disk operation
2662	9826	Track # for USR(X) disk operation
2D73	11635	String output-Routine from BASIC-ROM. After calling this sub-routine a string can be displayed on the screen.
2F0A	12042	Location of 24 used by random access file calc. routines
A000-A037	40960-41015	BASIC exit jump table
A000-BFFF	40960-49151	BASIC in ROM
A038-A065	41016-41061	BASIC non-initial word jumps (real entry addresses)
A084-A163	41092-41315	BASIC Keyboard in ASCII
A164-A186	41316-41350	Error messages
A1A1	41377	Look back thru stack ???
A212	41490	Check for OM and stack overflow
A24C	41548	"OM" error

A24E	41550	Error; caller sets X register to error code
A274	41588	BASIC Warm-start. To jump to this address is about the same as hitting W after BREAK. Prints "OK" on the screen
A357	41815	Input and fill buffer put null at end
A386	41862	Input from FFEB
A339	41881	Toggle CTRL O flag
A432	42034	Find BASIC line whose # is in 11, 12; put addr of ptr of that line in AA,AB
A477	42103	Call this routine and then jump to A5C2 and you'll be RUNing the current BASIC program - starting from machine language!
A491	42129	Clear stack; 0 in 8C and 61
A5C2	42434	Top of main BASIC execute loop
A5FC	42492	Entry to BASIC execute loop
A5FF	42495	Do line of BASIC
A629	42537	Jmp FFF1 for CTRL C
A636	42550	CTRL C entry point
A67B	42619	Set null count at D0 (?)
A77F	42879	This routine takes an expression whose starting address is 00C3 and 00C4 and changes from ACSCII-HEX to binary. The result is in the

		address 00AC, 00AD, 00AE, 00AF
A866	43110	Put null at end of buffer; CRLF; nulls
A86C	43116	CRLF w/nulls from 0D
A8C3	43203	Print routine for messages. Upper address in the Y-register, lower address in the A-register, ASCII-string
A8E0	43232	Output""
A8E3	43235	Output "?"
A8E5	43237	Output char in A; update 0E; check line length
A925	43301	Input routine less clear CTRL O
A946	43334	Output "?"; jump to A357
AAAD	43693	Get 16 bit arg from BASIC line; AE05 will put value in AE, AF; this is a TM error check
AAC1	43713	Like AAAD no type mismatch check.
ABA0	43936	Put0 in 5F; get char; goto B887 if numeric ???
ABD8	43992	16 bit complement using AE05/AFC1 ?
ABF5	44021	Checks for "(", call AAC1, checks for ")"
ABF5-AC0C	44021-44044	This series of routines (actually of entry points to one routine) uses the BC routine to check for various delimiters. If you

disassemble the ROM here, it will demonstrate classic use of the 2C opcode as a combination NOP and immediate load, depending on where you jump in. ABFB checks for ")";ABFE for "("; AC01 for ","; AC03 for whatever character you leave in A when you call it. ABF5 checks for "(", Calls AAC1 to get a value, then checks for ")".

(Thoughts of a BASIC statement X=USR(Y)(Z) should be jumping into your head about now.)

ABFB	44027	SN err if next char not ")"
ABFE	44030	SN err if next char not "("
AC01	44033	SN err if next char not ","
AC03	44035	SN err if next not what's in A
AC0C	44044	SN err printer
AD0B	44299	Get var name from BASIC line; put addr of var in 95, 96 and A,Y
AD53	44371	Expects var name in 93,94; finds addr of var and put in 95,96 and A,Y; 0 in 61
AE05	44549	INVAR puts 15 bit signed value in AE,AF

AE85	44677	BS error
AE88	44680	FC error
AFC1	44993	OUTVAR 0 in 5F;(A) in AE;(Y) in AF; then to ?
B0AE	45230	Msg printer (A8C3)
B3AE	45998	Put 8 bit arg from line in AE,AF
B3F3	46067	(BA,BB) to C3,C4
B408	46088	Changes a binary value in the floating point accumulator to a two byte number and puts it into locations 0011 and 0012
B4D0	46288	Arith to normalize FP args??
B887	47239	Check for +, -, $, #, ., E...long!
B95A	47450	Prints current line number
B95E	47454	Changes the hex number whose value is in the registers A and X, to a decimal number and displays it on the screen
B962	47458	Prints contents of AD,AE (as dec)
BD11	48401	BASIC cold-start address
BE4E	48718	"Written by" message
BEE4	48868	UART input routine (S1883 chip at FB0X)
BEF3	48883	UART output routine
BEFE	48894	UART initialization
BF07	48903	ACIA input (6850 chip at FC0X-like CII-4P)
BF15	48917	ACIA output routine
BF22	48930	ACIA initialization

BF2D	48941	Display routine. Displays contents of accumulator (ASCII value) on the screen
C704	50948	Data register for Port 1A
C705	50949	Control register for Port 1A
C706	50950	Data register for Port 1B
C707	50951	Control register for Port 1B
C708	50952	Data register for Port 2A
C709	50953	Control register for Port 2A
C70A	50954	Data register for Port 2B
C70B	50955	Control register for Port 2B
C70C	50956	Data register for Port 3A
C70D	50957	Control register for Port 3A
C70E	59058	Data register for Port 3B
C70F	59059	Control register for Port 3B
D000-D3FF	53248-54271	Picture repeat memory
D000-D7FF	53248-55295	Video Memory
DE00	56832	Tone Generator, character/line & Color on/off (0-7)
DF00	57088	Polled Keyboard
DF01	57089	Frequency for Tone Generator
E3EC	58348	Specifies Color (0-15)
F000-F001	61440-61441	Cassette Port 6850 (ACIA C1P)

F800	63488	Telephone Interface
F800-FFFF	63488-63535	Monitor EpROM
F801	63489	Telephone Interface Control Register for 63488
F802	63490	Telephone Interface
F803	63491	Telephone Interface Control Register for 63490
F804	63492	1PA4 MODEM self test 1PA5 MODEM squelch 1PA6 MODEM originate mc. 1PA7 MODEM answer mode 1CA1 DTWP decodes strobe
F805	63493	Control Register for 63492
F806	63494	1PB0-1PB7 Dialer Data (1CB1 ,1CB2 not used)
F807	63459	Control Register for 63494
F808	63496	ACIA
F809	63497	Control Register for 63496
FB00	64256	430 board's A/D converter output
FC00	64512	Floppy Initializing routine
FC01	64513	ACIA Data Register (for printer or modem)
FCB1	64689	Output from a byte of the A-register to the cassette
FD00	64768	Input of a character from the keyboard. The ASCII code of this character is put to the accumulator (A-register). Stored in addresses

		0200-0216 hex
FE00	65024	Entry to monitor. Erases screen, puts ACIA back
FE0C	65036	Entry to monitor without stack initialization
FE43	65091	Brings monitor into the address mode
FE80	65152	Input of an ASCII-character form cassette. Value is then in the A-register
FE93	65171	Changes an ASCII-hex number to its binary equivalent. Result in the A-register (80 if mistake)
FF00	65280	RESET entry point
FF69	65385	BASIC-routine for program-output on cassette. Puts a character to tape & displays it at the same time on the screen. For CR it outputs 10 nulls. The save flag has to be set (0205 = 1)
FF98	65435	Control-C routine
FFBA	65466	BASIC Input routine

GAMES

Archery

ARCHERY GAME - This is a game of skill--or luck!! It contains unique trajectory plot not often found in this type of game.

```
 100 REM C1P AND C2P VERSION OF ARCHERY.BY
DAVE WILKIE
 110 GOSUB1140
 120 INPUT"C1P OR C2P?????";T$
 130 IFT$="C1"THEN1120
 140 IFT$="C2"THEN1130
 150 GOTO120
 160 REM WE GOSUB TO READ VIDEO PARAMETERS
 170 REMC=LOWERLEFTSCREEN,DH=WIDTHOFSCREEN,
DV=HEIGHT,DD=ADDRESSINCREMENT
 180 REM BETWEEN VERTICALLY ADJACENT SCREEN
 CHARACTERS
 190 GOTO220
 200 FORI=1TO34:PRINT:NEXT
 210 RETURN
 220 GOSUB1140
 230 PRINT"      ARCHERY"
 240 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
 250 PRINT"   BE CAREFUL!!"
 260 PRINT" DON'T SHOOT YOURSELF"
 270 FORI=0TO1000:NEXT
 280 GOSUB1140
 290 PRINT"WHAT IS YOUR PROFICIENCY"
 300 INPUT"(1=BEGINNER) (2=AMATEUR) (3=PRO)
 (4=SUPER PRO)";PR
 310 GOSUB1140
 320 IFPR=1THENPRINT"            AW SHUCKS"
 330 IFPR=2THENPRINT"SO WHOSE FAULT IS THAT
?"
 340 IFPR=3THENPRINT"PROS ARE A DIME A DOZE
N"
 350 IFPR=4THENPRINT"WILLIAM TELL REINCARNA
TE":PRINT:PRINT"   WE'LL SOO
 360 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
 370 FORIC=1TO2000:NEXT
 380 GOSUB1140
 390 N=24/PR
 400 REM ENTER HERE FOLLOWING HIT
 410 R=100*PR*RND(1+PR)+50
 420 HAX=1.5*R:G=32.2
 430 REM ENTER HERE FOLLOWING MISS
 440 PRINT"YOU HAVE"N"ARROWS"
 450 PRINT"RANGE IS"R"FEET"
 460 PRINT"CHOOSE INITIAL VELOCITY"
 470 INPUT"LESS THAN 200ft/s";VO
 480 IFVO>200THENGOTO460
 490 PRINT"CHOOSE INITIAL ANGLE"
 500 INPUT"LESS THAN 90";TH
 510 IFTH>90THENGOTO500
 520 TH=TH/57.3
 530 REM VAX IS GOING TO BE MAXIMUM DISPLAY
 HEIGHT IN FEET. IT MAY BE
 540 REM   A CONSTANT OR THE EQUATION OF LI
NE 600
 550 GOTO570:REM THIS AVOIDS THE VERT. SCRE
EN FILL OPTION IN LINE 600.
 560 VAX=VO^2*(SIN(TH))^2/(2*G)
 570 VAX=R
 580 XD=2*SIN(TH)*COS(TH)*(VO^2)/G
 590 XE=XD-R
 600 XF=ABS(XE)
 610 GOSUB1140
 620 N=N-1
 630 FORI=0TODH:POKE(C+I),132
 640 NEXTI
 650 FORI=0TODV:POKE(C-DD*I),139:NEXTI
 660 POKE(C+INT(DH/1.5)),233
 670 FORI=0 TO DH:X=I*HAX/DH
 680 Y=X*TAN(TH)-(X^2)*G/(2*(VO*COS(TH))^2)
 690 YI=INT(Y*DV/VAX)*DD
 700 PK=C+I-YI
 710 IFPK>C+DDTHENGOTO860
 720 IFI=0THENCR=17
 730 IFI=0THENGOTO790
 740 IFYI>YLTHENCR=17
 750 IFYI>YLTHENGOTO790
 760 IFYI=YLTHENCR=18
 770 IFYI=YLTHENGOTO790
 780 CR=19
 790 POKE(PK),(CR)
 800 YL=YI
 810 NEXTI
 820 FORI=0TO1000:NEXTI
 830 PRINT
 840 PRINT"GEE WHIZ, EASE UP!"
 850 GOTO980
 860 FORII=0TO1000:NEXTII:IFXF>1/PRTHENGOTO
980
 870 FORTT=1TO2
 880 FORTI=0TO255:POKE(C+INT(DH/1.5)),TI:NE
XT
 890 POKE(C+INT(DH/1.5)),32
 900 NEXT TT
 910 PRINT
 920 PRINT" CONGRATULATIONS!!!":PRINT"
YOU HIT IT!!"
 930 REM GOOD SPOT FOR BULLS-EYE SCORING?
 940 H=H+1
 950 IFN=0GOTO1060
 960 PRINT:PRINT"***** NEW TARGET *****":PR
INT
 970 GOTO400
 980 IFXE<0THENXE$="UNDERSHOOT":XE=ABS(XE):
GOTO1000
 990 IFXE>0THENXE$="OVERSHOOT"
 1000 IFXD<30THENPRINT"FACT IS, YOU SHOT YO
UR-  SELF IN THE FOOT!"
 1010 IFXD<30THEN PRINT"BE MORE CAREFUL!":G
OTO1030
 1020 PRINTXE" FOOT "XE$
 1030 PRINT"VELOCITY WAS"VO:PRINT"ANGLE WAS
   "TH*57.3
 1040 IFN=0GOTO1060
 1050 GOTO440
 1060 PRINT"YOU'RE OUT OF ARROWS.   YOU HIT
"H"OUT OF"24/PR"     TIMES"
 1070 PRINT"WOULD YOU LIKE TO PLAY     AGAI
N?"
 1080 INPUTAG$
 1090 IFLEFT$(AG$,1)="Y"THEN GOTO350
 1100 PRINT"SOME WILLIAM TELL YOU   ARE"
 1110 END
 1120 C=54051:DH=25:DV=22:DD=32:GOTO160
 1130 C=55040:DH=63:DV=32:DD=64:GOTO160
 1140 FORX=1TO35:PRINT:NEXT
 1150 RETURN
OK
```

Ayatollah

AYATOLLAH - This game lets you attempt to rule Iran as the Ayatollah. Let's see what kind of a ruler you will be. Be sure to take good care of your hostages. Good luck!

```
 100 REM AYATOLLAH
 110 FORX1=0TO35:PRINT:NEXT
 120 PRINT"TEST YOUR ADMINISTRATIVE ABILITY
 BY RULING A THIRD ";
 130 PRINT"WORLD NATION
 140 PRINT"SUCCESSFULLY FOR A 10-YR. TERM O
F OFFICE.":PRINT
 150 D1=0:P1=0:Z=0:ZZ=0:Q=1
 160 Y=INT(RND(3)*20):A=INT(600*RND(0)+600)
:E=INT(RND(0)*200)
 170 I=INT(RND(0)*10):P=INT(RND(45)*140)
 180 IFY<3THENY=3
 190 IFI<1THENI=1
 200 IFP<65THENP=65
 210 H=Y*A:S=H-E
 220 BB=S/P
 230 AA=A/P
 240 D=0
 250 PRINT
 260 PRINT:PRINT:PRINT"AYATOLLAH: I AM TELL
ING YOU,":Z=Z+1
 270 PRINT"IN THE YEAR"Z","D"SLAVES STARVED
,"I"HOSTAGES ";
 280 PRINT"WERE CAPTURED
 290 P=P+I
 300 IFQ>0THEN350
 310 X=INT(RND(2.5)*1.5)+1.1
 320 P=INT(P/X)
 330 P=INT(P/2)
 340 PRINT"ANOTHER REVOLUTION OF RELIGIOUS
FANATICS. MANY SLAVES DIED
 350 PRINT"POPULATION IS NOW"P
 360 PRINT"THE CITY NOW OWNS"A"ACRES."
 370 PRINT"YOU HAVESTED"Y"BUSHELS PER ACRE.
"
 380 PRINT"THIEVERY ACCOUNTED FOR"E"BUSHELS
 390 E1=INT(E/4):IFE=0THENE1=435
 400 PRINT"TAXES WERE"E1"BUSHELS
 410 PRINT"YOU NOW HAVE"S"BUSHELS IN STORE.
":PRINT
 420 IFZ=51ANDZZ=4THEN1220
 430 IFZ=41ANDZZ=3THEN1220
 440 IFZ=31ANDZZ=2THEN1220
 450 IFZ=21ANDZZ=1THEN1220
 460 IFZ=11ANDZZ=0THEN1220
 470 C=INT(10*RND(1)):Y=C+17
 480 PRINT"LAND IS TRADING AT"Y"BUSHELS PER
 ACRE
 490 PRINT"HOW MANY ACRES DO YOU WISH TO BU
Y ";
 500 INPUTQ:IFQ<0THEN1190
 510 IFY*Q<=STHEN540
 520 GOSUB1090
 530 GOTO490
 540 IFQ=0THEN570
 550 A=A+Q:S=S-Y*Q:C=0
 560 GOTO630
 570 PRINT:PRINT"HOW MANY ACRES DO YOU WISH
 TO SELL ";
 580 INPUTQ:IFQ<0THEN1190
 590 IFQ<ATHEN620
 600 GOSUB1130
 610 GOSUB570
 620 A=A-Q:S=S+Y*Q:C=0
 630 PRINT
 640 PRINT"HOW MANY BUSHELS DO YOU WISH TO
FEED YOUR SLAVES";
 650 INPUTQ
 660 IFQ<=0THEN1190
 670 IFQ<=STHEN700
 680 GOSUB1090
 690 GOTO640
 700 S=S-Q:C=1:PRINT
 710 PRINT"HOW MANY ACRES DO YOU WISH TO PL
ANT ";
 720 INPUTD:IFD=0THEN850
 730 IFD<0THEN1190
 740 IFD<=ATHEN770
 750 GOSUB1130
 760 GOTO710
 770 IFINT(D/2)<STHEN800
 780 GOSUB1090
 790 GOTO710
 800 IFD<=10*PTHEN840
 810 PRINT"BUT YOU HAVE ONLY"P"SLAVES TO TE
ND THE FIELDS. NOW THEN,
 820 GOTO710
 830 S=S-INT(D/2)
 840 S=S-INT(D/2)
 850 GOSUB1170
 860 Y=C:H=D*Y:E=0
 870 GOSUB1170
 880 IFINT(C/2)<>C/2THEN900
 890 E=INT(S/C)
 900 S=S-E+H
 910 GOSUB1170
 920 I=INT(C*(20*A+S)/P/100+1)
 930 C=INT(Q/20)
 940 Q=INT(10*(RND(2)-.3)):Y=C+17
 950 IFP<CTHEN240
 960 D=P-C:IFD>.45*PTHEN990
 970 P1=((Z-1)*P1+D*100/P)/Z
 980 P=C:D1=D1+D:GOTO260
 990 FORX=0TO35:PRINT:NEXT
 1000 PRINT:PRINT:PRINT:PRINT
 1010 PRINT:PRINT"YOU STARVED"D"SLAVES IN O
NE YEAR!!!
```

```
 1020 PRINT
 1030 PRINT"DUE TO THIS HORRIBLE STUPIDITY,
 YOU NOT ONLY HAVE BEEN
 1040 PRINT"PROMOTED TO A HIGHER OFFICE AND
 HONORED BY YOUR PEERS,
 1050 PRINT
 1060 PRINT"BUT YOU HAVE BEEN UNANIMOUSLY E
LECTED 'SUPREME AYATOLLAH'"
 1070 GOTO1530
 1080 PRINT:PRINT
 1090 PRINT:PRINT
 1100 PRINT"AYATOLLAH-CANT YOU THINK."
 1110 PRINT"YOU HAVE ONLY"S"BUSHELS OF GRAI
N. NOW THEN,"
 1120 RETURN
 1130 PRINT:PRINT
 1140 PRINT"AYATOLLAH-DONKEY: TRY TO THINK"
;
 1150 PRINT". YOU ONLY HAVE"A"ACRES.
 1160 RETURN
 1170 C=INT(5*(RND(0)+1))
 1180 RETURN
 1190 PRINT:PRINT"AYATOLLA-WRETCH: I CANNOT
 STAND YOU ANY LONGER
 1200 PRINT"GET YOURSELF ANOTHER SUCKER. I
QUIT!!!!!!
 1210 GOTO1510
 1220 PRINT"IN YOUR"Z-1"YE R TERM IN OFFICE
 "P1"PERCENT OF THE
 1230 PRINT"POPULATION STARVED PER YEAR ON
AVERAGE, I.E.,A TOTAL OF"
 1240 PRINTD1"SLAVES DIED!!!":L1=A/P:CC=S/P
:L=L1+(CC/26)
 1250 PRINT"YOU STARTED WITH"AA"ACRES PER S
LAVE AND ENDED WITH
 1260 PRINTL1"ACRES PER SLAVE.
 1270 PRINT"YOU STARTED WITH"BB"BUSHELS PER
 SLAVE AND ENDED WITH":PRINT
 1280 PRINTCC"BUSHELS PER SLAVE.":PRINT
 1290 FORT=0TO20000:NEXT
 1300 IFP1>33THEN1000
 1310 IFL<7THEN1000
 1320 IFP1>10THEN1420
 1330 IFL<9THEN1420
 1340 IFP1>3THEN1460
 1350 IFL<10THEN1460
 1360 FORT=0TO35:PRINT:NEXT
 1370 PRINT:PRINTTAB(15)"CONGRATULATIONS":P
RINT
 1380 PRINT"AFTER CHANGING YOUR RELIGION YO
U BECAME HUMAN
 1390 PRINT:PRINTTAB(15)"*****************"
:PRINT
 1400 PRINT:PRINTTA2(10)"YOU WERE A FANTAST
IC LEADER
 1410 PRINT:PRINT:GOTO1510
 1420 PRINT"YOUR HEAVY-HANDED PERFORMANCE S
MACKS OR NERO AND SOME"
 1430 PRINT"OTHER TERRIBLE PEOPLE
 1440 PRINT"THE SLAVES (REMAINING) FIND YOU
 AN UNPLEASANT RULER AND,
 1450 PRINT"FRANKLY, HATE YOUR GUTS!":GOTO1
510
 1460 PRINT"YOUR PERFORMANCE COULD HAVE BEE
N SOMEWHAT BETTER,BUT"
 1470 PRINT"REALLY WASN'T TOO BAD AT ALL. "
;
 1480 PRINTINT(P*(RND(.8)))"SLAVES WOULD
 1490 PRINT"DEARLY LIKE TO SEE YOU ASSASSIN
ATED BUT WE ALL HAVE OUR"
 1500 PRINT"TRIVIAL PROBLEMS."
 1510 PRINT
 1520 PRINT
 1530 END
OK
```

Ball Dance

BALL DANCE - This graphics program causes a number of balls to "dance" around the screen. The program demonstrates:

1) *Animation - balls move around the screen.*
2) *Programmability - you can "program" each ball to make certain moves.*
3) *Multidimensionality - there can be many balls, each with its own set of moves. To do this, the program uses a three dimensional array.*

The program first asks you for the number of balls in the dance. Then for each ball it asks:

A) *the number of steps in the cycle for this ball's dance*
B) *the horizontal and vertical displacements for each of the dance steps.*

The program will then display the dance of the balls.

PROGRAM EXPLANATION:

LINE	*DESCRIPTION*
20	*Clears screen*
30	*Initializes variables for screen*
50-60	*Inputs number of balls and*

dimensions array accordingly

70-150 Loops through for each ball

70 Establishes random starting postion (P(I))

90-100 Inputs number of movements

110 Loops for each movement

130-140 Inputs horizontal and vertical displacement

160 Clears screen

170-240 Loops for each ball, goes to next step (C(I)=C(I)+1)

180 If last step in dance return to step 1

190 Finds new position from formula:
New position=old position + horizontal step + (vertical step * width)

200-210 If new position is out of the screen, go back to top or bottom of screen

NOTE: These checks are very important. They prevent the program from destroying itself.

220 POKES new ball position and blanks out old position

230 Sets old ball position equalto the new position for next time through loop

250 Goes back to another display of the balls

VARIABLES:

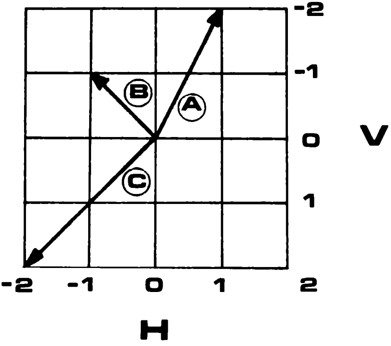
C(I) = Actual index for ball I

P(I) = Actual position for ball I

S(I,J,1) = Horizontal movement for ball I, movement J

S(I,J,2) = Vertical movement for ball I, movement J

FOR EXAMPLE:



Step (A) is 1,2
Step (B) is -1,-1
Step (C) is -2,2

IF you want the ball to stay in its place,you may enter 0,0.

When all the steps for all the balls have been entered, the program will begin the dance.

```
 10 REM BALL DANCE
 20 FORX=1TO30:PRINT:NEXT
 30 LO=53318:HI=54236:WID=32
 40 L=HI-LO:TK=226
 50 INPUT"HOW MANY BALLS";N
 60 DIM S(N,10,2),T(N),P(N),C(N)
 70 FORI=1TON:P(I)=INT(RND(1)*L)+LO
 80 PRINT
 90 PRINT"HOW MANY MOVEMENTS FOR BALL #";I
 100 INPUTT(I)
 110 FORJ=1TOT(I)
 120 PRINT
 130 PRINT"ENTER H AND V FOR BALL";I;"MOVEM
ENT";J
 140 INPUTS(I,J,1),S(I,J,2)
 150 NEXTJ,I
 160 FORX=1TO30:PRINT:NEXT
 170 FORI=1TON:C(I)=C(I)+1
 180 IFC(I)>T(I)THENC(I)=1
 190 NP=P(I)+S(I,C(I),1)+WID*S(I,C(I),2)
 200 IFNP>HITHENNP=NP-L
 210 IFNP<LOTHENNP=NP+L
 220 POKENP,TK:POKEP(I),32
 230 P(I)=NP
 240 NEXTI
 250 GOTO170
```

Black Box

THE BLACK BOX - In this game,the player hunts for a number of atoms randomly placed in a 2-dimensional, 9"x9", lattice. The program provides a mobile cursor which responds to keys 1 through 4 by moving up, down, and back and forth, respectively. With this cursor, the player is able to hunt for these atoms with a light-beam. Key 5 provides the light-beam. The player must beware of multiple reflections.

```
10 CLEAR
20 FORQ=1TO30:PRINT:NEXT
30 PRINT"  WELCOME"
```

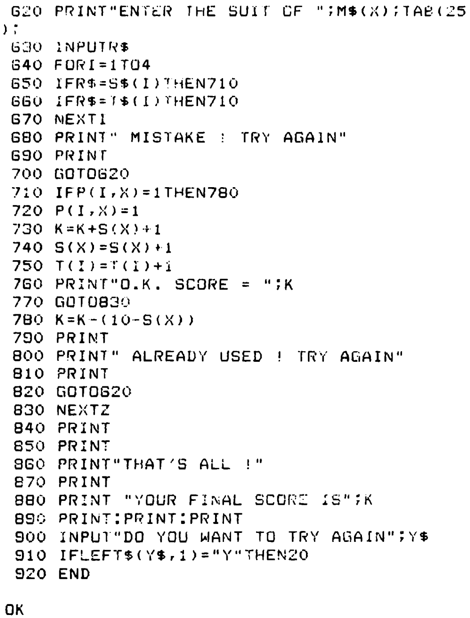
```
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
50 FORI=1TO1500:NEXT
60 FORQ=1TO30:PRINT:NEXT
70 PRINT"  CALCULATING"
80 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
90 DIMA(10,10)
100 FORI=0TO10:FORJ=0TO10:A(J,I)=0
110 IFI=0ORI=10ORJ=0ORJ=10THENA(J,I)=2
120 NEXTJ:NEXTI
130 N=1
140 FORQ=1TO30:PRINT:NEXT
150 INPUT"HOW MANY ATOMS SHOULD I HIDE";X
160 IFX<81THEN180
170 PRINT"X SMALLER 81 PLEASE":GOTO150
180 FORI=1TOX
190 K=INT(RND(10)*9+1):J=INT(RND(20)*9+1)
200 IFN=2THEN230
210 IFJ=1ORJ=9ORK=1ORK=9THENN=N+1:GOTO190
220 IFJ=10ORK=10THEN190
230 IFA(J,K)=9THEN190
240 A(J,K)=9
250 NEXTI
260 FORQ=1TO30:PRINT:NEXT
270 S=53341
280 FE=54287
290 ST=64
300 Q=0
310 FORI=STOFESTEPST
320 N=0
330 IFQ=0ORQ=2ORQ=12ORQ=14THEN400
340 FORJ=10TO26STEP2
350 IFQ=1ORQ=13THENPOKE(I+J),49+N:N=N+1:GO
TO370
360 POKEI+J,91:POKEI+J+1,93
370 NEXTJ
380 IFQ<3ORQ>11THEN400
390 POKEI+7,46+Q:POKEI+30,46+Q
400 Q=Q+1
410 IFQ=15THENQ=0
420 NEXTI
430 POKE530,1
440 MO=0:AL=0:QT=32:QU=32:HT=2
450 PL=53477:HN=53477
```

```
 460  POKE57088,127:TE=PEEK(57088)
 470  IFHT=0THENPOKE53434,161
 480  IFHT=2THENPOKE53434,32
 490  IFTE=127THENMO=MO-1
 500  IFTE=191THENMO=MO+1
 510  IFTE=223THENAL=AL-1
 520  IFTE=239THENAL=AL+1
 530  IFTE=247THEN640
 540  IFMO<0ORMO>10THENMO=5:GOTO460
 550  IFAL<0ORAL>10THENAL=0
 560  ZX=PL+64*MO+2*AL
 570  IFPEEK(ZX)=91ANDHT=2THENMO=10:AL=10
 580  POKEHN,QT:POKEHN+1,QU
 590  HN=PL+MO*64+2*AL
 600  QT=PEEK(HN):QU=PEEK(HN+1)
 610  POKEHN,62:POKEHN+1,60
 620  FORI=1TO200:NEXTI
 630  GOTO460
 640  IFMO=0ANDAL=10THEN460
 650  IFMO=10ANDAL=10THEN460
 660  IFMO=0ANDAL=0THEN850
 670  IFMO=10ANDAL=0THEN960
 680  IFHT=0THEN880
 690  IN=PL+MO*64+AL*2:POKEIN,5:POKEIN+1,7
 700  A(MO,AL)=7
 710  IFAL=10THENJ=MO:K=9:C=0:D=-1
 720  IFAL=0THENJ=MO:K=1:C=0:D=1
 730  IFMO=0THENJ=1:K=AL:C=1:D=0
 740  IFMO=10THENJ=9:K=AL:C=-1:D=0
 750  N=6
 760  IFA(J,K)=9ORA(J,K)=90THENGOSUB1140 :GO
TO830
 770  IFA(J,K)=2THENA(J,K)=7:GOSUB1150 :GOTO
830
 780  U=ABS(D):V=ABS(C)
 790  IFA(J+U,K+V)=9ORA(J+U,K+V)=90THENJ=J-C
:K=K-D:C=-U:D=-V
 800  IFA(J-U,K-V)=9ORA(J-U,K-V)=90THENJ=J-C
:K=K-D:C=U:D=V
 810  IFA(J,K)=7THENPOKEIN,5:POKEIN+1,7:GOTO
830
 820  J=J+C:K=K+D:GOTO760
 830  MO=0:AL=0
 840 GOTO590
 850 IFHT=0THENHT=2:GOTO460
 860 HT=0
 870 GOTO460
 880 IFMO=0ORMO=10ORAL=0ORAL=10THEN460
 890 POKE53498,181
 900 P7=PL+MO*64+AL*2
 910 R5=QT
 920 IFR5=42THEN1100
 930 POKEP7,42:POKEP7+1,42:A(MO,AL)=A(MO,AL
)*10
 940 GOTO590
 950 WR=0
 960 FORI=1TO9:FORJ=1TO9
 970 VN=PL+I*64+J*2
 980 IFPEEK(VN)=42THENYZ=YZ+1
 990 IFA(I,J)=90THENWR=WR+1:POKEVN,161:POKE
VN+1,161
 1000 IFA(I,J)=9THENPOKEVN,161:POKEVN+1,161
 1010 NEXTJ:NEXTI
 1020 IFYZ-X>0THENPRINT"CHEAT...":GOTO1050
 1030 IFWR=XTHENPRINT"WELL DONE ":GOTO1050
 1040 PRINT"HARD LUCK,YOU HAD ";X-WR;" WRON
G."
 1050 INPUT"WANT AGAIN";W$
 1060 YZ=0
 1070 IFW$="Y"THEN10
 1080 POKE530,0
 1090 END
 1100 A(MO,AL)=A(MO,AL)/10
 1110 POKEP7,91:POKEP7+1,93
 1120 POKE53576,32
 1130 GOTO590
 1140 POKEIN,187:POKEIN+1,187:RETURN
 1150 EX=PL+64*J+K*2
 1160 FORI=1TO10
 1170 POKEEX,32:POKEEX+1,32
 1180 FORO=1TO50:NEXTO
 1190 POKEEX,5:POKEEX+1,7
 1200 FORO=1TO50:NEXTO
 1210 NEXTI
 1220 RETURN
OK
```

Concentration

CONCENTRATION - A card game that tests your ability to concentrate. You are attempting to choose each of the cards in a deck without duplication. If you choose the same card twice, you lose points.

```
 10 REM CONCENTRATION
 20 FORX=1TO30:PRINT:NEXT
 30 PRINT"IMAGINE YOU HAVE A PACK OF CARDS.
"
 40 PRINT"PICK A CARD"
 50 PRINT"EITHER OF A CERTAIN SUIT"
 60 PRINT" OR OF A CERTAIN VALUE."
 70 PRINT
 80 PRINT"IF YOU CHOOSE A NEW CARD YOU SCOR
E MORE POINTS,"
 90 PRINT "BUT IF YOU TRY TO"
 100 PRINT"SELECT A CARD THAT'S ALREADY USE
D"
 110 PRINT"YOU LOSE POINTS."
 120 PRINT
 130 PRINT
 140 DIM P(4,13),N$(13),S$(4),T(4),S(13),M$
(13),T$(4)
 150 FORI = 1 TO 4
 160 FORJ = 1 TO 13
 170 P(I,J)=0
 180 NEXTJ
 190 NEXTI
 200 FORI=1TO13
 210 READ N$(I),M$(I)
 220 S(I)=0
 230 NEXTI
 240 FORI=1TO4
 250 READ S$(I),T$(I)
 260 T(I)=0
 270 NEXTI
 280 DATA "ACE", "AN ACE","TWO","A TWO","TH
REE","A THREE","FOUR"
 290 DATA "A FOUR","FIVE","A FIVE","SIX","A
 SIX","SEVEN","A SEVEN"
 300 DATA "EIGHT","AN EIGHT","NINE","A NINE
","TEN","A TEN","JACK"
 310 DATA "A JACK","QUEEN","A QUEEN","KING"
,"A KING"
 320 DATA "CLUB","CLUBS","DIAMOND","DIAMOND
S"
 330 DATA "HEART","HEARTS","SPADE","SPADES"
 340 FORZ=1TO52
 350 X=INT(103*RND(1)+1)
 360 IFX<52THEN600
 370 X=X-4*INT(X/4)+1
 380 IFT(X)>12THEN370
 390 PRINT"ENTER A ";S$(X);TAB(25);
 400 INPUT R$
 410 FORI=1TO13
 420 IFR$=N$(I)THEN480
 430 IFR$=M$(I)THEN480
 440 NEXTI
 450 PRINT" MISTAKE ! TRY AGAIN"
 460 PRINT
 470 GOTO390
 480 IFP(X,I)=1THEN550
 490 P(X,I)=1
 500 K=K+T(X)+1
 510 S(I)=S(I)+1
 520 T(X)=T(X)+1
 530 PRINT"O.K. SCORE = ";K
 540 GOTO830
 550 K=K-(20-T(X))
 560 PRINT
 570 PRINT" ALREADY USED ! TRY AGAIN"
 580 PRINT
 590 GOTO390
 600 X=X+1-13*(INT(X/13))
 610 IFS(X)>3THEN600
```



```
 620 PRINT"ENTER THE SUIT OF ";M$(X);TAB(25
);
 630 INPUTR$
 640 FORI=1TO4
 650 IFR$=S$(I)THEN710
 660 IFR$=T$(I)THEN710
 670 NEXTI
 680 PRINT" MISTAKE : TRY AGAIN"
 690 PRINT
 700 GOTO620
 710 IFP(I,X)=1THEN780
 720 P(I,X)=1
 730 K=K+S(X)+1
 740 S(X)=S(X)+1
 750 T(I)=T(I)+1
 760 PRINT"O.K. SCORE = ";K
 770 GOTO830
 780 K=K-(10-S(X))
 790 PRINT
 800 PRINT" ALREADY USED ! TRY AGAIN"
 810 PRINT
 820 GOTO620
 830 NEXTZ
 840 PRINT
 850 PRINT
 860 PRINT"THAT'S ALL !"
 870 PRINT
 880 PRINT "YOUR FINAL SCORE IS";K
 890 PRINT:PRINT:PRINT
 900 INPUT"DO YOU WANT TO TRY AGAIN";Y$
 910 IFLEFT$(Y$,1)="Y"THEN20
 920 END

OK
```

Magic Square

MAGIC SQUARE - In this game you put numbers in a square, then arrange them so that the sum of each row, column, and diagonal, equals 15.

```
 10 REM MAGIC SQUARE
 20 FORX=1TO30:PRINT:NEXT
 30 PRINT"INSTRUCTIONS: THE PLAYER SELECTS
A NON-USED NUMBER"
 40 PRINT"BETWEEN 1 AND 9 AND PUTS IT IN AN
 EMPTY FIELD OF THE"
 50 PRINT"MAGIC SQUARE.":PRINT
 60 PRINT"THE OBJECT OF THE GAME IS TO ARRA
NGE THE NUMBERS SUCH"
 70 PRINT"THAT THE SUM OF EACH ROW, COLUMN,
 AND DIAGONAL IS 15."
 80 PRINT"THE PLAYER THAT SETS THE FIRST NU
MBER GIVING A WRONG"
 90 PRINT"TOTAL, LOSES.":PRINT
 100 PRINT"THE COMPUTER WILL ASK YOU EACH T
IME, IN WHICH POSITION"
 110 PRINT"YOU WANT THE SELECTED NUMBER.  E
NTER '2,5' IF YOU WANT"
 120 PRINT"TO PUT A 5 IN POSITION 2.  THE P
OSITIONS ARE NUMBERED"
 130 PRINT"FROM TOP LEFT TO BOTTOM RIGHT: 1
23, 456, 789."
 140 CLEAR:GOSUB600
 150 PRINT:INPUT"YOUR ROUND ! ----LOCATION
AND NUMBER";I,N
 160 IFI<1ORI>9ORN<1ORN>9THEN180
 170 IFA(I)=0ANDB(N)=0THEN190
 180 PRINT:PRINT"WRONG ENTRY!...TRY AGAIN":
GOTO150
 190 A(I)=N:B(N)=1:M=M+1
 200 GOSUB600
```

```
210 GOSUB450
220 IFW=0THEN240
230 PRINT:PRINT"SORRY,YOU LOST!":GOTO420
240 IFM<5THEN260
250 PRINT"FFFFF":GOTO420
260 FORQ=1TO9
270 IFA(Q)>0THEN350
280 FORR=1TO9
290 IFB(R)>0THEN340
300 A(Q)=R
310 GOSUB450
320 IFW=0THEN370
330 Q1=Q:R1=R:W=0:A(Q)=0
340 NEXTR
350 NEXTQ
360 W=1:R=R1:Q=Q1:A(Q)=R
370 B(R)=1
380 PRINT:PRINT"I PUT A"R"IN POSITION"Q
390 GOSUB600
400 IFW=0THEN150
410 PRINT"I'VE LOST ---- YOU'VE WON!"
420 INPUT" DO YOU WANT TO PLAY AGAIN ";Y$
430 IFLEFT$(Y$,1)="Y"THEN90
440 IFLEFT$(Y$,1)="N"THEN END
450 FORX=1TO8
460 ONXGOTO470,480,490,500,510,520,530
470 J=1:K=2:L=3:GOTO540
480 K=4:L=7:GOTO540
490 K=5:L=9:GOTO540
500 J=4:L=6:GOTO540
510 J=3:L=7:GOTO540
520 K=6:L=9:GOTO540
530 J=7:K=8:GOTO540
540 IFA(J)=0ORA(K)=0ORA(L)=0THEN560
550 IFA(J)+A(K)+A(L)<>15THEN580
560 NEXTX
570 GOTO590
580 W=1
590 RETURN
600 PRINT:PRINTTAB(25)A(1);A(2);A(3)
610 PRINTTAB(25)A(4);A(5);A(6):PRINTTAB(25
)A(7);A(8);A(9):PRINT:RETURN
620 END
```

Mickey Mouse

MICKEY MOUSE - Take a guess! This is a picture of your favorite Disney character, Mickey Mouse!

```
 10 FOR T=1 TO 30:PRINT:NEXT
 20 PRINT"THIS PROGRAM DRAWS MICKEY MOUSE"
 30 FOR T=1 TO 2500:NEXT
 40 FOR T=1 TO 30:PRINT:NEXT
 50 PRINT"=================================
===========================
 60 PRINT"                *******
   *******"
 70 PRINT"               *********
  *********"
 80 PRINT"              ***********
 ***********"
 90 PRINT"             *************
*************"
 100 PRINT"             *************
 *************"
 110 PRINT"             *******************
**************"
 120 PRINT"             ************.... ..
..************"
 130 PRINT"             ***********..  ...
 ..***********"
 140 PRINT"              ******* **..  ...
 ..** *******"
 150 PRINT"                      **.. *...*
 ..**"
 160 PRINT"                      **.....-..
...**"
 170 PRINT"                    *.......***.
......*"
 180 PRINT"                   *........***.
```

```
.......*"
 190 PRINT"                  *.............
........*"
 200 PRINT"                  *......***...*
**......*"
 210 PRINT"                  *......*******
**......*"
 220 PRINT"                   *......**+++*
*......*"
 230 PRINT"                    *......**+**
......*"
 240 PRINT"                      *......*..
....*"
 250 PRINT"                        *.......
..*"
 260 PRINT"                           *...*
"
 270 PRINT"                            ***"
 280 PRINT"================================
==========================="
 290 END
OK
```

Space Shuttle

SPACE SHUTTLE - You are attempting to land on earth in a spaceship. Fuel, velocity, and distance are given and the player must determine the amount of thrust necessary for a smooth landing. After each "?" you must type in a thrust command number between 0 and 500.

```
 10 REM SPACE SHUTTLE
 20 FORX=0TO35:PRINT:NEXT
 30 PRINT"*********************************
**********************************
 40 PRINT:PRINT
 50 PRINT"YOU HAVE JUST DEPARTED A SATELLIT
E"
 60 PRINT"AND ARE DESCENDING BACK TO EARTH"
 70 PRINT"AFTER EACH ? TYPE IN A ROCKET THR
UST
 80 PRINT"COMMAND AS A NUMBER BETWEEN 0 AND
 500
 90 PRINT"UNITS(EACH UNIT=200 NEWTONS=900 L
BS. OF THRUST
 100 PRINT"----------GOOD LUCK----------
 110 PRINT:PRINT
 120 PRINT"********************************
*****************************
 130 PRINT:PRINT:PRINT:PRINT
 140 PRINT
 150 PRINT"TIME:   VELOCITY:  ALTITUDE:  FU
EL:    THRUST:
 160 PRINT"SECONDS METERS/SEC KILOMETERS KI
LOGRAMS NEWTON/100
 170 PRINT
 180 H=110
 190 H=INT((H-70)*RND(1)+70)
 200 S=5000
 210 E=10000
 220 T=0
 230 G=2.4
 240 GOTO260
 250 V=-100
 260 PRINTT;TAB(8);V;TAB(19);H;TAB(30);E;TA
B(38);"F=";
 270 INPUTF
 280 IF ABS(F-250)>251THEN 310
 290 IFF>ETHEN330
 300 GOTO350
 310 PRINT"ILLEGAL THRUST--PLEASE REPEAT
 320 GOTO260
 330 PRINT"ONLY";E;"UNITS OF FUEL LEFT--PLE
ASE REPEAT
 340 GOTO260
 350 E=E-F
 360 V1=V
 370 V2=2000*F/(S+E)-10*I
 380 V=V+V2
 390 D=(V1+V2/2)*10
 400 H=H+D/1000
 410 IFH<=.01THEN500
 420 IFE<=0THEN450
 430 T=T+10
 440 GOTO260
 450 PRINT"YOU ARE OUT OF FUEL";H;"KILOMETE
RS UP
 460 PRINT"      IT WAS NICE KNOWING YOU
 470 PRINT:PRINT
 480 GOTO600
 490 F=0
 500 K=0
 510 K=K+.1
 520 V1=1
 530 V2=20*F/(S+E)-.1*I
 540 V=V+V2
 550 H=H-(V1+V2/2)/1000
 560 PRINT"IMPACT IN APPROXIMATELY";K;"SECO
NDS
```

```
 570 PRINT"STANDBY":PRINT:PRINT
 580 IFV>=-5THEN640
 590 IFV>=-15THEN670
 600 PRINT"CRUNCH!!! VE1OCITY="V;"M/SEC
 610 PRINT"THAT'S";V*3.2*3600/5280;"MPH--OH
 BOY!
 620 PRINT"TURN IN YOUR LICENSE AT ONCE
 630 END
 640 PRINT"BEAUTIFUL LANDING V=";V*3.2*3600
/5280;"MPH
 650 PRINT"YOU HAD";E;"KILOGRAMS OF FUEL LE
FT
 660 END
 670 PRINT"CLUNK--RATTLE--RATTLE--SQUINCH
 680 PRINT"V= ";V*3.2*3600/5280;"MPH: ROUGH
,BUT YOU SURVIVED
 690 END

OK
```

Tank in a Trap

TANK IN A TRAP - This program moves a tank around within a walled square. There are two escape routes. See if the tank can find them.

```
 10 REM TANK IN A TRAP
 20 FORI=1TO30:PRINT:NEXT
 30 POKE57000,0
 40 O=53568:U=55040:POKEO,204:POKEO+31,205:
POKEU,203
 50 POKEU+31,206:FORI=O+1TOO+30:POKEI,131:N
EXT
 60 FORI=O+64TOU-32STEP64:POKEI,140:NEXT
 70 FORI=O+95TOU-1STEP64:POKEI,139:NEXT
 80 FORI=U+1TOU+30:POKEI,132:NEXT
 90 FORC=0TO29
 100 FORI=O+65+CTOU-62+CSTEP64:POKEI,161:NE
XTI:NEXTC
 110 W=1000
 120 P=INT(54033+10*RND(2)):D=1
 130 FORQ=1TO50:NEXT
 140 GOSUB170:IFP=53634ORP=55006THEN350
 150 W=W-1:IFW<1THEN90
 160 GOTO130
 170 ONDGOTO220,180,260,300
 180 S=PEEK(P+63):IFS=140THEND=4:RETURN
 190 IFS=203THEND=1:RETURN
 200 IFS=132THEND=3:RETURN
 210 POKEP,32:POKEP+63,253:P=P+63:RETURN
 220 S=PEEK(P-63):IFS=131THEND=4:RETURN
 230 IFS=205THEND=3:RETURN
 240 IFS=139THEND=3:RETURN
 250 POKEP,32:POKEP-63,249:P=P-63:RETURN
```

```
260 S=PEEK(P-65):IFS=131THEND=2:RETURN
270 IFS=204THEND=4:RETURN
280 IFS=140THEND=1:RETURN
290 POKEP,32:POKEP-65,255:P=P-65:RETURN
300 S=PEEK(P+65):IFS=132THEND=1:RETURN
310 IFS=206THEND=3:RETURN
320 IFS=139THEND=2:RETURN
330 POKEP,32:POKEP+65,251:P=P+65:RETURN
340 PRINT:PRINT:PRINT
350 PRINT" T A N K  E S C A P E D"
360 FORI=1TO1000:NEXT
370 GOTO10

OK
```

Turnabout

TURNABOUT - In this game, you get N numbers (N is defined in line 40) in a random sequence. To win, you must arrange the numbers in numerical order. This is accomplished by interchanging certain parts of the list.

Example:

The List:

5 4 3 2 1 6 7

Can be changed to:

1 2 3 4 5 6 7

by answering the question: "HOW MANY SHALL I TURN?" with: 5.

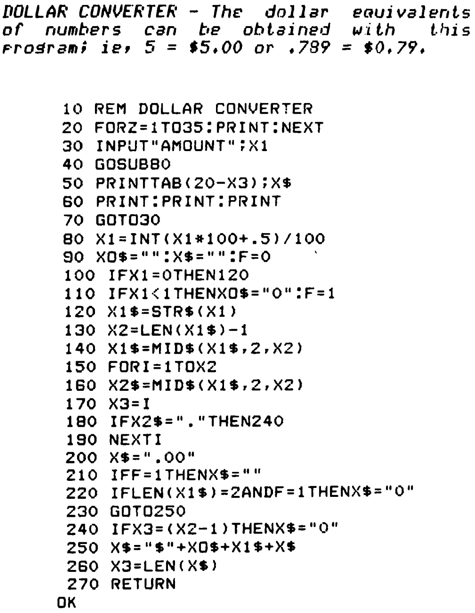
```
10 REM TURNABOUT
20 FORQ=1TO30:PRINT:NEXT
30 DIM A(20)
40 N=7:REM CAN BE CHANGED
50 A(1)=INT((N-1)*RND(1))+2
60 FOR K=2 TO N
70 A(K)=INT(N*RND(1))+1
80 FOR L=1 TO K-1
90 IF A(K)=A(L) THEN 70
100 NEXT L
110 NEXT K
120 PRINT
130 PRINT"THE LIST LOOKS AS FOLLOWS :"
```

```
 140 PRINT
 150 T=0
 160 GOSUB 360
 170 PRINT
 180 INPUT"HOW MANY SHALL I TURN";R
 190 IF R<=N THEN 220
 200 PRINT:PRINT"MAXIMUM IS";N;" !!!"
 210 GOTO 170
 220 T=T+1
 230 FOR K=1 TO INT(R/2)
 240 Z=A(K)
 250 A(K)=A(R-K+1)
 260 A(R-K+1)=Z
 270 NEXT K
 280 GOSUB 360
 290 FOR K=1 TO N
 300 IF A(K)<>K THEN 170
 310 NEXT K
 320 PRINT:PRINT:PRINT
 330 PRINT"YOU NEEDED"T"MOVES !"
 340 PRINT:PRINT:PRINT
 350 STOP
 360 PRINT
 370 FOR L=1 TO N
 380 PRINT A(L);
 390 NEXT L
 400 PRINT
 410 PRINT
 420 RETURN
 430 END

OK
```

PERSONAL UTILITIES

Dollar Converter
Calorie Counter
Speed vs. Gasoline Consumption
Gasoline Consumption vs. Speed
German Vocabulary
Astrology
Intra-Ocular Lens Power

Dollar Converter



DOLLAR CONVERTER - The dollar equivalents of numbers can be obtained with this program; ie, 5 = $5.00 or .789 = $0.79.

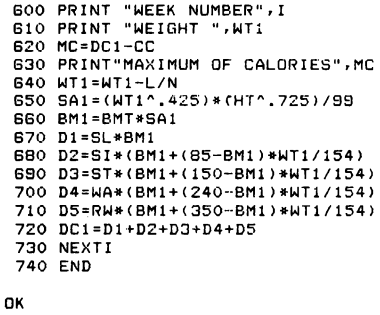
```
10 REM DOLLAR CONVERTER
20 FORZ=1TO35:PRINT:NEXT
30 INPUT"AMOUNT";X1
40 GOSUB80
50 PRINTTAB(20-X3);X$
60 PRINT:PRINT:PRINT
70 GOTO30
80 X1=INT(X1*100+.5)/100
90 X0$="":X$="":F=0
100 IFX1=0THEN120
110 IFX1<1THENX0$="0":F=1
120 X1$=STR$(X1)
130 X2=LEN(X1$)-1
140 X1$=MID$(X1$,2,X2)
150 FORI=1TOX2
160 X2$=MID$(X1$,2,X2)
170 X3=I
180 IFX2$="."THEN240
190 NEXTI
200 X$=".00"
210 IFF=1THENX$=""
220 IFLEN(X1$)=2ANDF=1THENX$="0"
230 GOTO250
240 IFX3=(X2-1)THENX$="0"
250 X$="$"+X0$+X1$+X$
260 X3=LEN(X$)
270 RETURN
OK
```

Calorie Counter

CALORIE COUNTER - This exercise calculates the maximum daily caloric intake to lose a certain amount of weight according to sex, height, weight, and amount of daily exercise.

```
 10 REM CALORIE CALCULATIONS
 20 FORQ=1TO30:PRINT:NEXT
 30 INPUT"IS SUBJECT MALE OR FEMALE (M OR F
)";S$
 40 IFS$="M"THEN80
 50 M=-.0808
 60 B=37.5769
 70 GOTO100
 80 M=-.0846
 90 B=39.8
 100 INPUT"ENTER HEIGHT (INCHES)";HT
 110 INPUT"ENTER AGE";AG
 120 INPUT"ENTER WEIGHT (POUNDS)";WT
 130 GOTO150
 140 PRINT "HOURS DO NOT EQUAL 24, TRY AGAI
N"
 150 PRINT "INPUT HRS/DAY FOR EACH OF THE F
OLLOWING ACTIVITIES"
 160 INPUT"SLEEPING";SL
 170 INPUT"SITTING";SI
 180 INPUT"STANDING";ST
 190 INPUT"WALKING";WA
 200 INPUT"RAPID WALKING";RW
 210 T=SL+SI+ST+WA+RW
 220 IFT<>24THEN140
 230 INPUT"ENTER DESIRED WEIGHT LOSS IN POU
NDS";L
```

```
240 INPUT"ENTER # OF WEEKS ";N
250 PRINT:PRINT:PRINT:PRINT:PRINT
260 SA=(WT^.425)*(HT^.725)/99
270 BMT=AG*M+B
280 BM=BMT*SA
290 DC=0
300 C1=SL*BM
310 C2=SI*(BM+(85-BM)*WT/154)
320 C3=ST*(BM+(150-BM)*WT/154)
330 C4=WA*(BM+(250-BM)*WT/154)
340 C5=RW*(BM+(350-BM)*WT/154)
350 DC=C1+C2+C3+C4+C5
360 CC=L/N*500
370 IFS$="M"THENPRINT"MALE"
380 IFS$="F"THENPRINT"FEMALE"
390 PRINT "HEIGHT(IN)=",HT
400 PRINT "WEIGHT(#)=",WT
410 PRINT"AGE=",AG
420 PRINT
430 PRINT "HRS/DAY SLEEPING=",SL,"CALORIES
=",C1
440 PRINT "HRS/DAY SITTING=",SI,"CALORIES=
",C2
450 PRINT "HRS/DAY STANDING",ST,"CALORIES=
",C3
460 PRINT "HRS/DAY WALKING",WA,"CALORIES="
,C4
470 PRINT"HRS/DAY RAPID WALKING",RW,"CALOR
IES=",C5
480 PRINT
490 PRINT"DAILY CALORIES TO MAINTAIN WEIGH
T:",DC
500 PRINT
510 PRINT "DESIRED WEIGHT LOSS(#)=",L
520 PRINT "NUMBER OF WEEKS=",N
530 PRINT "CHANGE IN CALORIES PER DAY=",CC
540 PRINT
550 WT1=WT:DC1=DC
560 PRINT:PRINT:PRINT
570 INPUT"DO YOU WISH THE WEEKLY SCHEDULE
(Y/N)";Z$
580 IFZ$="N"THENEND
590 FORI=1 TO N
```



```
 600 PRINT "WEEK NUMBER",I
 610 PRINT "WEIGHT ",WT1
 620 MC=DC1-CC
 630 PRINT"MAXIMUM OF CALORIES",MC
 640 WT1=WT1-L/N
 650 SA1=(WT1^.425)*(HT^.725)/99
 660 BM1=BMT*SA1
 670 D1=SL*BM1
 680 D2=SI*(BM1+(85-BM1)*WT1/154)
 690 D3=ST*(BM1+(150-BM1)*WT1/154)
 700 D4=WA*(BM1+(240-BM1)*WT1/154)
 710 D5=RW*(BM1+(350-BM1)*WT1/154)
 720 DC1=D1+D2+D3+D4+D5
 730 NEXTI
 740 END

OK
```

Speed vs. Gasoline Consumption

SPEED VERSUS GASOLINE CONSUMPTION - If you want to know the total cost per mile of operating your automobile, try this one. Enter the price of gasoline, price of the automobile, cost of insurance, registration fees, and extra costs per year. The program calculates the total cost per mile.

```
 10 REM COSTS A MILE
 20 FORQ=1TO30:PRINT:NEXT
 30 C$="CHECK WHAT YOU ENTERED"
 40 PRINT"TOTAL COSTS A MILE"
 50 PRINT
 60 INPUT"AMOUNT AT LAST REFUEL";A
 70 INPUT"MILAGE INDICATION AT LAST REFUEL"
;K1
 80 INPUT"MILAGE INDICATION AT NEXT TO LAST
 REFUEL";K2
 90 IFK2>=K1THENPRINTC$:PRINT:GOTO70
 100 INPUT"PRICE OF THE CAR";P
 110 INPUT"ESTIMATED WORKING LIFE (YEARS)";
L
 120 IFLEN(STR$(L))>2THENPRINTC$:PRINT:GOTO
110
 130 INPUT"NORMAL INTEREST RATE (%/YEAR)";I
R
 140 INPUT"AVERAGE DISTANCE A YEAR";D
 150 INPUT"INSURANCE A YEAR";IN
 160 INPUT"TAX A YEAR";T
 170 INPUT"ESTIMATED EXTRA COSTS A YEAR";E
 180 PRINT
 190 A1=A/(K1-K2)
 200 A2=(P/L)+(P/100*IR)+IN+T+E
 210 A3=A2/D
 220 A4=A1+A3
 230 FORQ=1TO30:PRINT:NEXT
 240 PRINT"TOTAL COSTS A MILE ARE :"
 250 PRINT
 260 A4=INT(A4*100+.5)/100
 270 PRINTTAB(10);A4;"$"
 280 PRINT:PRINT:PRINT
 290 INPUT"ENTER THE DISTANCE YOU WANT TO D
RIVE";DI
 300 PRINT:PRINT:PRINT:PRINT:PRINT
 310 PRINT"THIS TRIP COSTS YOU";A4*DI;"$"
 320 END
OK
```

Gasoline Consumption vs. Speed

GASOLINE CONSUMPTION VERSUS SPEED - If you wish to know the amount of gasoline per mile which your automobile consumes; this program will calculate it, taking into consideration the weight, speed, wind resistance on different kinds of vehicles, and tire resistance on the road.

```
10 REM GAS CONSUMPTION
20 FORQ=1TO30:PRINT:NEXT
30 DEFFNA(X)=INT(100*X+.5)/100
40 PRINT"DATES OF YOUR CAR"
50 PRINT
60 INPUT"WEIGHT (POUNDS)";G
70 G=G*.4536
80 INPUT"WIDTH (FEET)";B
90 B=B*.3048
100 INPUT"HEIGHT (FEET)";H
110 H=H*.3048
120 PRINT:PRINT
130 PRINT"FORM OF YOUR CAR,ENTER :"
140 PRINT"1,IF TRUCK"
150 PRINT"2,IF BUS"
160 PRINT"3,IF FULL SIZE OR SMALL"
170 PRINT"4,IF COUPE"
180 PRINT"5,IF STATION WAGON"
190 INPUTF
200 PRINT:PRINT:PRINT
210 PRINT"DATES OF YOUR TRIP"
220 PRINT
230 INPUT"HOW MANY MILES";D
240 D=D*1.6093
250 INPUT"WHICH SPEED (MI/HR)";V
```

```
 260 V=V*1.6093
 270 INPUT"HOW MANY PASSENGERS";P
 280 INPUT"HOW MUCH BAGGAGE (POUNDS)";L
 290 L=L*.4536
 300 PRINT:PRINT:PRINT
 310 PRINT:PRINT:PRINT
 320 G=G+P*75+L
 330 A=B*H*.9
 340 DATA.8,.6,.4,.3,.5
 350 FORQ=1TOF
 360 READC
 370 NEXTQ
 380 R=G*.015
 390 W=C*.0048*A*V/3.6*V/3.6
 400 B=(R+W)/1.5*D/100*.8
 410 PRINT:PRINT:PRINT
 420 PRINT"RESULTS :"
 430 PRINT
 440 PRINT"TIRE FRICTION=";FNA(R*2.2046);"P
OUNDS"
 450 PRINT"AIR RESISTANCE=";FNA(W*2.2046);"
POUNDS"
 460 PRINT"AT";FNA(V*.6214);"MILES/HOUR"
 470 PRINT
 480 PRINT"FOR";FNA(D*.6214);"MILES"
 490 PRINT"YOUR CAR WILL CONSUME";FNA(B*.26
42);"GAL"
 500 PRINT"THAT IS";FNA(D/B*2.352);"MI/GAL"
 510 T=D/V
 520 H=INT(T)
 530 M=INT((T-H)*60+.5)
 540 PRINT
 550 PRINT"DRIVING TIME=";H;"HOURS";M;"MIN.
"
 560 PRINT
 570 INPUT"DO YOU WANT TO TRY ANOTHER SPEED
 (Y/N)";T$
 580 IFT$="N"THEN630
 590 PRINT
 600 INPUT"WHICH SPEED (MI/HR)";V
 610 V=V*1.6093
 620 PRINT:PRINT:GOTO390
 630 PRINT"O.K. HAPPY MOTORING !"
 640 END
```

German Vocabulary

GERMAN VOCABULARY - With this program you can learn German vocabulary. As you learn the words already in the program, you can enter new words to be learned and store them on tape.

```
 20 FORX=1TO30:PRINT:NEXT
 30 T=5:S=FRE(1)/200
 40 DIM E$(T,S),D$(T,S)
 50 READ N(1),N(2),N(3),N(4),N(5)
 60 FORK=1TO5
 70 FORI=1TON(K)
 80 READ E$(K,I),D$(K,I):NEXTI:NEXTK
 90 PRINT"DO YOU WANT:":PRINT
 100 PRINT:PRINT"G - GERMAN-ENGLISH EXERCIS
E"
 110 PRINT:PRINT"N - ENTER NEW WORDS"
 120 PRINT:PRINT"S - STORE ON TAPE"
 130 INPUTW$
 140 IFW$="N"THEN170
 150 IFW$="G"THEN260
 160 IFW$="S"THEN350
 170 PRINT:PRINT:PRINT:PRINT:N(1)=N(1)+1
 180 IFN(1)>STHENPRINT"MEMORY FULL":N(1)=N(
1)-1:GOTO90
 190 PRINT "FIRST THE ENGLISH WORD"
 200 PRINT:INPUT E$(1,N(1))
 210 IFE$(1,N(1))="N"THENN(1)=N(1)-1:PRINT:
GOTO90
 220 PRINT:PRINT
 230 PRINT"AND NOW THE GERMAN":PRINT
 240 INPUT D$(1,N(1))
 250 PRINT:PRINT:GOTO170
 260 Z=-1
 270 PRINT:PRINT:PRINT:PRINT:Z=Z+1:IFZ=10TH
ENGOSUB450
 280 GOSUB630
 290 PRINT" ";D$(M1,N(M1));"=";TAB(18);
 300 INPUTA$
 310 IFA$="N"THEN90
 320 IFA$=E$(M1,N(M1))THENPRINT:GOSUB520:GO
TO270
 330 PRINT:PRINT"THE CORRECT WORD IS:",TAB(
20)
 340 PRINTE$(M1,N(M1)):PRINT:PRINT:GOSUB570
:GOTO270
 350 POKE57000,1:SAVE
 360 PRINT" 1001 ";"DATA";
 370 FORI=1TO4:PRINTN(I);",";:NEXT:PRINTN(5
)
 380 T$=",":Z=1002:FOR K=1 TO 5
 390 FOR I=1 TO N(K) STEP 2
 400 PRINTZ;"DATA";E$(K,I);T$;D$(K,I);
 410 IFE$(K,I+1)=""THENPRINTT$;E$(K,I+1);T$
;D$(K,I+1):GOTO430
 420 PRINT
 430 Z=Z+1:NEXTI:NEXTK
 440 LIST-660
 450 FORI=1TO5:PRINTTAB(6*I-4);N(I);:NEXT
 460 FOR I=1 TO 10:PRINT:NEXT
 470 IFN(1)>5THENPRINT"  ENTER NEW WORDS!"
 480 PRINT:PRINT:PRINT
 490 IFB=0THENPRINT" ALL 10 RIGHT,VERY GOOD
":GOTO510
 500 PRINT"   ";J;"RIGHT";B;" WRONG":PRINT:
PRINT
 510 PRINT:PRINT:GOSUB660:Z=0:J=0:B=0:RETUR
N
 520 J=J+1:IF M1=5 THEN N(M1)=N(M1)-1:RETUR
N
 530 FOR I=N(M1+1) TO 1 STEP -1
 540 E$(M1+1,I+1)=E$(M1+1,I):D$(M1+1,I+1)=D
$(M1+1,I):NEXT
```

```
 550 E$(M1+1,1)=E$(M1,N(M1)):D$(M1+1,1)=D$(
M1,N(M1))
 560 N(M1+1)=N(M1+1)+1:N(M1)=N(M1)-1:RETURN
 570 B=B+1:A$=E$(M1,N(M1)):B$=D$(M1,N(M1))
 580 FOR I=N(1) TO 1 STEP -1
 590 E$(1,I+1)=E$(1,I):D$(1,I+1)=D$(1,I):NE
XT
 600 E$(1,1)=A$:D$(1,1)=B$
 610 N(1)=N(1)+1:N(M1)=N(M1)-1:RETURN
 620 END
 630 M=0:FOR I=1 TO 5
 640 IF N(I)>M THEN M=N(I):M1=I
 650 NEXT I:RETURN
 660 FORI=1TO5000:NEXT
 670 DATA  6 , 2 , 3 , 3 , 3
 680 DATA ALMOST,FAST,HASTE,HAST
 690 DATA SHODDY,WERTLOS PROTZIG,WORK,ARBEI
T
 700 DATA RUN,LAUFEN,NEARBY,NAHE BEI
 710 DATA STUBBY,UNTERSETZT DICK ,DITCH,GRA
BEN
 720 DATA WHIMPER,WINSELN,NOOK,SCHLUPFWINKE
L
 730 DATA CALF,KALB
 740 DATA BUSH,BUSCH,HOST,GASTGERBER
 750 DATA OVEN,BACKOFEN,ALMOST,FAST
 760 DATA SWAP,TAUSCHEN,YOLK,EIDOTTER
 770 DATA BULLET,KUGEL,GLOVE,HANDSCHUH

OK
```

ASTROLOGY:

CALCULATION OF MC AND AS - The positions of MC and AS are the two most important things in an exact, time-and-place referenced, horoscope.

MC means "Medium Coeli", or Medium of the Heaven. It is the location of the ecliptic, which is at the local meridian at a certain time.

AS means "The Ascending". It is the location of the ecliptic, which is rising at the Astrological East horizon at the same time.

ARMC means "Ascensio Recta Medium Coeli", the location of the heavenly equator, which is at the local meridian at that time.

Input: B = Latitude: Enter 46.1322 for 46
            degrees 13' 22"
        R$ = N for Northern Latitude; S for
            Southern Latitude
         G = Time of Birth: Enter 18.13.07
         for 18h 13m 7s

Formulas:
   For MC position:
      tan R = tan AR/cos E
   Declination of MC:
      sinD = sin MC*sin E

AS position:

*AS = tan(to the minus 1)(cos AR*sin E*tan(Latitude-Dmc)+MC+90)*

```
 10 REM CALCULATION OF MC AND AS
 20 FORQ=1TO30:PRINT:NEXT
 30 PRINT"POSITIONS IN DEGREES MIN SEC"
 40 PRINT
 50 PI=3.1415927
 60 U=PI/180:T=180/PI
 70 SE=SIN(23.45*U):CE=COS(23.45*U)
 80 INPUTB,R$,G1
 90 FORQ=1TO30:PRINT:NEXT
 100 PRINT"LATITUDE DMS ";R$;B
 110 GOTO210
 120 REM CONVERSION TO DEGREES
 130 X=X+1E-6:X1=(X-INT(X))*100:X2=(X1-INT(
X1))/.6
 140 X3=(INT(X1)+X2)/60+INT(X):X4=INT(X3*1E
4)+.5
 150 X=INT(X4)/1E4
 160 RETURN
 170 Y1=(Y-INT(Y))*60:Y2=((Y1-INT(Y1))*.6)+
INT(Y1)
 180 Y3=(INT(Y)+Y2/100)*1E4+.5
 190 Y=INT(Y3)*1E-4
 200 RETURN
 210 X=B
 220 GOSUB130:B=X:IFR$="S"THENB=B*-1
 230 X=G1
 240 GOSUB130:G=X
 250 AR=G*15
 260 MC=ATN(TAN(AR*U)/CE)*T
 270 IF AR>270THENMC=MC+360
 280 IF AR>90ANDAR<=270THEN MC=MC+180
 290 D1=SIN(MC*U)*SE
 300 D=ATN(D1/SQR(1-D1^2))*T
 310 AS=ATN(COS(AR*U)*SE*TAN((B-D)*U))*T+MC
+90
 320 IF AS>=360THENAS=AS-360
 330 Y=MC
 340 GOSUB170:MC=Y
 350 Y=AS
 360 GOSUB170
 370 AS=Y
 380 PRINT
 390 PRINT"BIRIM HMS";G1;TAB(10)"MC.DMS";TA
B(20)"AS.DMS"
 400 PRINT
 410 PRINT"ARMC";SPC(2)"GDEZ";TAB(11)AR;TAB
(20)MC;TAB(29)AS
 420 END

OK
```

Intra-Ocular Lens Power

INTRA-OCULAR LENS POWER - The intra-ocular lens power required to to make any eye emmetropic is computed. Refractive error and aphakic refraction is computed and appropriate individual lens power based on biometric data for surgical procedures is selected. Calculations are based on a 4/3 value for the refractive index of the cornea.

```
 10 REM INTRAOCULAR LENS POWER
 20 FORX=1TO30:PRINT:NEXT
 30 INPUT"AVERAGE KERATOMETER READINGS (MM)
";K
 40 INPUT"AXIAL LENGTH (MM)";A
 50 INPUT"POSTOPERATIVE ANTERIOR CHAMBER DE
PTH (MM)";D
 60 DE=1336*(4*K-A)/((A-D)*(4*K-D))
 70 PRINT
 80 INPUT"SPECTACLE REFRACTION (DIOPTERS)";
SR
 90 INPUT"VERTEX DISTANCE (METERS)";V
 100 D1=1336*(4*K-A-(V*(4*K-A)+.003*K*A)*SR
)
 110 D2=(A-D)*(4*K-D-(V*(4*K-D)+.003*D*A)*S
R)
 120 IP=D1/D2
 130 PRINT
 140 INPUT"DIOTROPIC POWER OF THE LENS (AQU
EOUS)";DP
```

```
 150 T=1336*(4*K-A)-DP*(A-D)*(4*K-D)
 160 T1=1336*(V*(4*K-A)+.003*A*K)-DP*(A-D)*
(V*(4*K-D)+.003*D*K)
 170 RE=T/T1
 180 PRINT:PRINT:PRINT:PRINT:PRINT
 190 PRINT "AVERAGE OF KERATOMETER READINGS
 (MM)=";K
 200 PRINT"AXIAL LENGTH (MM)=";A
 210 PRINT"POSTOP.ANTERIOR CHAMBER DEPTH (M
M)=";D
 220 PRINT"D EMM=";DE
 230 PRINT "SPECTACLE REFRACTION IN DIOPTER
S=";SR
 240 PRINT "VERTEX DISTANCE (METERS)=";V
 250 PRINT"INTRAOC.LENS POWER REQ.TO CREATE
 REF.ERR.=";IP
 260 PRINT"DIOTR.POWER OF THE INTRAOC.LENS
(AQUEOUS)=";DP
 270 PRINT "REFRACTIVE ERROR=";RE
 280 END

OK
```

HINTS AND INSTRUCTIONS

Tape/Disk - Disk/Tape Transfer

TAPE/DISK and DISK/TAPE Transfer

To transfer a BASIC program from tape to disk (and reverse) on a Challenger C1P with OS65D V. 3 system, the following procedures will require a terminal width of 132 (normal, if no conversions have been made).

A. Tape to disk

1. Type NEW (CR)
2. Type NULL08 (CR)
3. Type DISK!"IO 01" (CR)
4. Start the recorder. (don't worry about garbage in the beginning)
5. Stop recorder
6. Type LIST (CR)
7. After the listing, type DISK!"PUT(file name)" (CR)

B. Disk to tape

1. Type DISK!"LOAD(File name)" (CR)
2. Type LIST#1 (no CR)
3. Turn recorder to record, then hit CR
4. Here you will get no listing, but an "OK" at the end of the record
5. Turn the recorder off.

Two Computer Interface RS-232 to RS-232

TWO COMPUTER INTERFACE - RS-232 to RS-232

Instructions

To dump from C2-8P MF to C2-4P Cassette:

1. On the C2-8P, type LIST#, (RS232 PORT number)EXAMPLE: LIST#,3
2. On the C2-4P, type LOAD (CR)
3. On the C2-8P MF, hit (CR)

To dump from C2-4P Cassette to C2-8P MF:

1. Create a file to dump to on the C2-8P MF
2. On the C2-8P MF, type NEW
3. On the C2-8P MF, type DISK!"IO(space)(RS-232 PORT number, terminal number)EXAMPLE: DISK!"IO 04,02(For the input port, use 2 digits ; for the output port, use 2 digits)
4. On the C2-4P, type SAVE (CR) LIST then type one space
5. On the C2-8P MF hit (CR)
6. On the C2-4P, hit (CR)
7. On the C2-8P MF, type Disk!"FU(space)(FILE NAME) EXAMPLE: DISK!FU

POKE and PEEK

POKE AND PEEK:

POKE - The operator stores an integer N in a location W of memory. An error is reported if the number to be stored is out of range. Programs that unintentionally POKE values into pages $00,01 or 02 can cause very peculiar errors as the run continues, eventually BASIC may become so scrambled that RESET must be done. Since variables that haven't been defined have a value of zero, address $0000 is often ruined. Then if the (BREAK) key is hit, a warm start cannot be accomplished. Of course, expressions can be arguments of POKE.

```
   Example: Y*4+17,X+3
```

PEEK(X)

This is a function, not a command. But it is the natural opposite of POKE. PEEK returns the value (as a decimal integer between 0 and 255 inclusive) of the contents of address W.

Example:

```
10 I=3
20 ?PEEK(I*256)

RUN
 0
```

Self-starting BASIC Program

SELF-STARTING BASIC PROGRAM - When you SAVE a program, enter at the end of the program, while the recorder is still recording:

Enter POKE 545,1:RUN<CR

Similarly, you can reset the SAVE flag:

Enter POKE 517,0 <CR>

STOP

STOP - STOP causes an exit from the immediate mode with the printing of a break message.

Example:

```
20 FOR I=1 TO 10
30 IF I=3 THEN STOP
40 NEXT

RUN
BREAK IN 30
OK
```

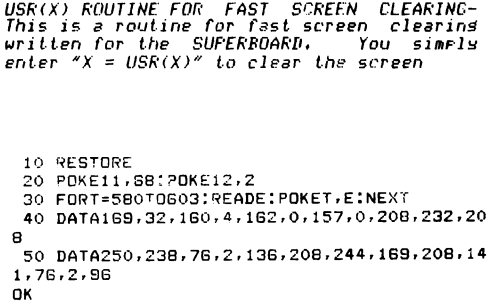
Important Tip

IMPORTANT TIP

There will be times when you inadvertently hit 'C' instead of 'W' after you hit the break key, or sometimes a warmstart is not possible. In some instances, it is possible to save your program:

```
   After "MEMORY SIZE?":
   Hit Break and "M"
   Hit "/" for DATA-mode
   Hit "CR",74,"CR",A2,"CR",C3,"CR" A8,"CR"
   Hit Break and "W"
   By this you change the addresses in 0001
   and 0004.
```

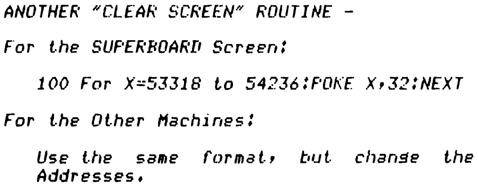
USR(X) for Fast Screen Clear



USR(X) ROUTINE FOR FAST SCREEN CLEARING-
This is a routine for fast screen clearing
written for the SUPERBOARD. You simply
enter "X = USR(X)" to clear the screen

```
 10 RESTORE
 20 POKE11,68:POKE12,2
 30 FORT=580TO603:READE:POKET,E:NEXT
 40 DATA169,32,160,4,162,0,157,0,208,232,20
8
 50 DATA250,238,76,2,136,208,244,169,208,14
1,76,2,96
OK
```

Another Fast Screen Clear



ANOTHER "CLEAR SCREEN" ROUTINE -

For the SUPERBOARD Screen:

```
   100 For X=53318 to 54236:POKE X,32:NEXT
```

For the Other Machines:

Use the same format, but change the Addresses.

USEFUL MATH ROUTINES

USEFUL MATH ROUTINES - This chapter contains 25 often-needed routines for mathematical functions. They can be used as subroutines or as stand-alone programs. The following programs are contained in this chapter:

1. *3/3 Determinant*
2. *4/4 Determinant*
3. *5/5 Determinant*
4. *Determinants of 3/3, 4/4, 5/5 Matrix*
5. *Matrix Multiplication*
6. *Classical Adjunct*
7. *Matrix Inversion*
8. *Peculiar Value of 3/3 Matrix*
9. *System of Linear Equations*
10. *Co-ordinant Transformation*
11. *Geometry*
12. *Calculation of PI*
13. *Number Converter*
14. *Sorting (Binary Tree)*
15. *Numerical Differentiation*
16. *Numerical Integration (Simpson)*
17. *Differential Equation*
18. *Prime Factors*
19. *Pythagorean Numbers*
20. *Decibel Program*
21. *Histograms*
22. *Regression Analysis*
23. *Simple Statistics*
24. *Function Plot*
25. *Precipitation*

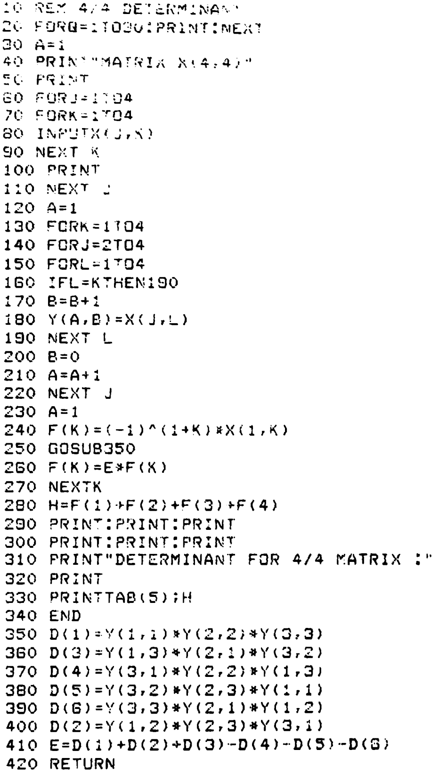
Determinant Programs

DETERMINANT PROGRAMS - The first four programs calculate determinants. The first, a 3 times 3; the second, a 4 times 4; the third, a 5 times 5; matrix. The fourth program gives the option of the size of the matrix. In all of these programs, the numbers of the matrix must be entered by lines.

1. 3/3 Determinant

```
10 REM 3/3 DETERMINANT
20 FORQ=1TO30:PRINT:NEXT
30 PRINT"MATRIX Y(3,3)"
40 PRINT
50 FORJ=1TO3
60 FORK=1TO3
70 INPUTY(J,K)
80 NEXT K
90 PRINT
100 NEXT J
110 D(1)=Y(1,1)*Y(2,2)*Y(3,3)
120 D(3)=Y(1,3)*Y(2,1)*Y(3,2)
130 D(4)=Y(3,1)*Y(2,2)*Y(1,3)
140 D(5)=Y(3,2)*Y(2,3)*Y(1,1)
150 D(6)=Y(3,3)*Y(2,1)*Y(1,2)
160 D(2)=Y(1,2)*Y(2,3)*Y(3,1)
170 E=D(1)+D(2)+D(3)-D(4)-D(5)-D(6)
180 PRINT
190 PRINT"DETERMINANT OF Y"
200 PRINT
210 PRINTE
220 END
```

2. 4/4 Determinant



```
10 REM 4/4 DETERMINANT
20 FORQ=1TO30:PRINT:NEXT
30 A=1
40 PRINT"MATRIX X(4,4)"
50 PRINT
60 FORJ=1TO4
70 FORK=1TO4
80 INPUTX(J,K)
90 NEXT K
100 PRINT
110 NEXT J
120 A=1
130 FORK=1TO4
140 FORJ=2TO4
150 FORL=1TO4
160 IFL=KTHEN190
170 B=B+1
180 Y(A,B)=X(J,L)
190 NEXT L
200 B=0
210 A=A+1
220 NEXT J
230 A=1
240 F(K)=(-1)^(1+K)*X(1,K)
250 GOSUB350
260 F(K)=E*F(K)
270 NEXTK
280 H=F(1)+F(2)+F(3)+F(4)
290 PRINT:PRINT:PRINT
300 PRINT:PRINT:PRINT
310 PRINT"DETERMINANT FOR 4/4 MATRIX :"
320 PRINT
330 PRINTTAB(5);H
340 END
350 D(1)=Y(1,1)*Y(2,2)*Y(3,3)
360 D(3)=Y(1,3)*Y(2,1)*Y(3,2)
370 D(4)=Y(3,1)*Y(2,2)*Y(1,3)
380 D(5)=Y(3,2)*Y(2,3)*Y(1,1)
390 D(6)=Y(3,3)*Y(2,1)*Y(1,2)
400 D(2)=Y(1,2)*Y(2,3)*Y(3,1)
410 E=D(1)+D(2)+D(3)-D(4)-D(5)-D(6)
420 RETURN
```

3. 5/5 Determinant

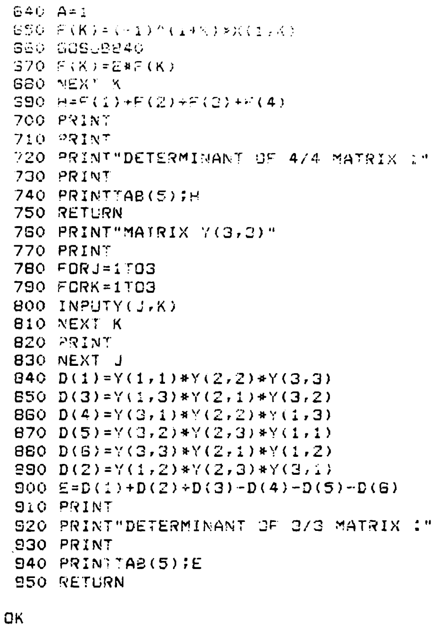
```
10 REM 5/5 DETERMINANT
20 FORG=1TO30:PRINT:NEXT
30 PRINT"MATRIX Z(5,5)"
40 PRINT
50 FORJ=1TO5
60 FORK=1TO5
70 INPUTZ(J,K)
80 NEXT K
90 PRINT
100 NEXT J
110 PRINT:PRINT:PRINT
120 PRINT"  CALCULATING..."
130 A=1
140 FORC=1TO5
150 FORJ=2TO5
160 FORL=1TO5
170 IFL=CTHEN200
180 B=B+1
190 X(A,B)=Z(J,L)
200 NEXT L
210 B=0
220 A=A+1
230 NEXT J
240 A=1
250 S(C)=(-1)^(1+C)*Z(1,C)
260 GOSUB340
270 S(C)=S(C)*H
280 NEXT C
290 P=S(1)+S(2)+S(3)+S(4)+S(5)
300 FORG=1TO30:PRINT:NEXT
310 PRINT"DETERMINANT OF 5/5 MATRIX :"
320 PRINT:PRINTTAB(5);P
330 END
340 A=1
350 FORK=1TO4
360 FORJ=2TO4
370 FORL=1TO4
380 IFL=KTHEN410
390 B=B+1
400 Y(A,B)=X(J,L)
410 NEXT L
```

```
420 B=0
430 A=A+1
440 NEXT J
450 A=1
460 F(K)=(-1)^(1+K)*X(1,K)
470 GOSUB520
480 F(K)=E*F(K)
490 NEXT K
500 H=F(1)+F(2)+F(3)+F(4)
510 RETURN
520 D(1)=Y(1,1)*Y(2,2)*Y(3,3)
530 D(3)=Y(1,3)*Y(2,1)*Y(3,2)
540 D(4)=Y(3,1)*Y(2,2)*Y(1,3)
550 D(5)=Y(3,2)*Y(2,3)*Y(1,1)
560 D(6)=Y(3,3)*Y(2,1)*Y(1,2)
570 D(2)=Y(1,2)*Y(2,3)*Y(3,1)
580 E=D(1)+D(2)+D(3)-D(4)-D(5)-D(6)
590 RETURN
```

4. *Determinants of 3/3, 4/4, 5/5 Matrix*

```
10 REM DETERMINANTS OF 3/3,4/4,5/5 MATRIX
20 FORG=1TO30:PRINT:NEXT
30 INPUT"ENTER SIZE OF MATRIX";A
40 PRINT:PRINT:PRINT
50 IFA=5THEN130
60 IFA=4THEN110
70 IFA=3THEN90
80 STOP
90 GOSUB760
100 STOP
110 GOSUB440
120 STOP
130 GOSUB160
140 STOP
150 PRINT
160 PRINT"MATRIX Z(5,5)"
170 FORJ=1TO5
180 FORK=1TO5
190 INPUTZ(J,K)
200 NEXT K
```

```
210 PRINT
220 NEXT J
230 A=1
240 FORC=1TO5
250 FORJ=2TO5
260 FORL=1TO5
270 IFL=CTHEN300
280 B=B+1
290 X(A,B)=Z(J,L)
300 NEXT L
310 B=0
320 A=A+1
330 NEXT J
340 A=1
350 S(C)=(-1)^(1+C)*Z(1,C)
360 GOSUB530
370 S(C)=S(C)*H
380 NEXT C
390 P=S(1)+S(2)+S(3)+S(4)+S(5)
400 PRINT
410 PRINT"DETERMINANT OF 5/5 MATRIX :"
420 PRINT:PRINTTAB(5);P
430 RETURN
440 A=1
450 PRINT"MATRIX X(4,4)"
460 PRINT
470 FORJ=1TO4
480 FORK=1TO4
490 INPUTX(J,K)
500 NEXT K
510 PRINT
520 NEXT J
530 A=1
540 FORK=1TO4
550 FORJ=2TO4
560 FORL=1TO4
570 IFL=KTHEN600
580 B=B+1
590 Y(A,B)=X(J,L)
600 NEXT L
610 B=0
620 A=A+1
630 NEXT J
```

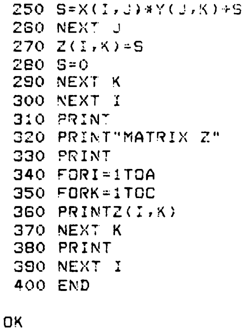


```
640 A=1
650 F(K)=(-1)^(1+K)*X(1,K)
660 GOSUB240
670 F(K)=Z*F(K)
680 NEXT K
690 H=F(1)+F(2)+F(3)+F(4)
700 PRINT
710 PRINT
720 PRINT"DETERMINANT OF 4/4 MATRIX :"
730 PRINT
740 PRINTTAB(5);H
750 RETURN
760 PRINT"MATRIX Y(3,3)"
770 PRINT
780 FORJ=1TO3
790 FORK=1TO3
800 INPUTY(J,K)
810 NEXT K
820 PRINT
830 NEXT J
840 D(1)=Y(1,1)*Y(2,2)*Y(3,3)
850 D(3)=Y(1,3)*Y(2,1)*Y(3,2)
860 D(4)=Y(3,1)*Y(2,2)*Y(1,3)
870 D(5)=Y(3,2)*Y(2,3)*Y(1,1)
880 D(6)=Y(3,3)*Y(2,1)*Y(1,2)
890 D(2)=Y(1,2)*Y(2,3)*Y(3,1)
900 E=D(1)+D(2)+D(3)-D(4)-D(5)-D(6)
910 PRINT
920 PRINT"DETERMINANT OF 3/3 MATRIX :"
930 PRINT
940 PRINTTAB(5);E
950 RETURN

OK
```

Matrix Multiplication

MATRIX MULTIPLICATION - In this program, you first have to enter the sizes of the two matrices and then the numbers of the matrices (one after the other) by lines. The program then will give Matrix Z--the product of the two matrixes which were entered.

```
 10 REM MATRIX MULTIPLICATION
 20 FORQ=1TO30:PRINT:NEXT
 30 INPUT"ENTER A,B,C FROM X(A,B) AND Y(B,C
)";A,B,C
 40 PRINT
 50 PRINT"MATRIX X"
 60 FORJ=1TOA
 70 FORK=1TOB
 80 INPUT X(J,K)
 90 NEXT K
 100 PRINT
 110 NEXT J
 120 PRINT
 130 PRINT"MATRIX Y"
 140 FORJ=1TOB
 150 FORK=1TOC
 160 INPUT Y(J,K)
 170 NEXT K
 180 PRINT
 190 NEXT J
 200 PRINT
 210 S=0
 220 FORI=1TOA
 230 FORK=1TOC
 240 FORJ=1TOC
```

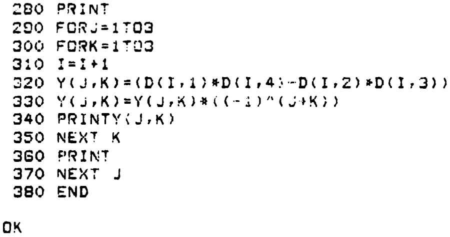


```
 250 S=X(I,J)*Y(J,K)+S
 260 NEXT J
 270 Z(I,K)=S
 280 S=0
 290 NEXT K
 300 NEXT I
 310 PRINT
 320 PRINT"MATRIX Z"
 330 PRINT
 340 FORI=1TOA
 350 FORK=1TOC
 360 PRINTZ(I,K)
 370 NEXT K
 380 PRINT
 390 NEXT I
 400 END

OK
```

Classical Adjunct

CLASSICAL ADJUNCT - This program gives all adjuncts of a previously entered 3 x 3 matrix.

```
10 REM CLASSICAL ADJUNCT
20 REM 3/3 MATRIX
30 FORQ=1TO30:PRINT:NEXT
40 PRINT"MATRIX X"
50 FORJ=1TO3
60 PRINT
70 FORK=1TO3
80 INPUTX(J,X)
90 NEXTK
100 NEXT J
110 F=0
120 FORJ=1TO3
130 FORK=1TO3
140 F=F+1
150 FORL=1TO3
160 FORM=1TO3
170 IFL=JORM=KTHEN200
180 E=E+1
190 D(F,E)=X(L,M)
200 NEXTM
210 NEXTL
220 E=0
230 NEXT K
240 NEXT J
250 I=0
260 PRINT:PRINT:PRINT
270 PRINT"ADJUNCT OF X"
```

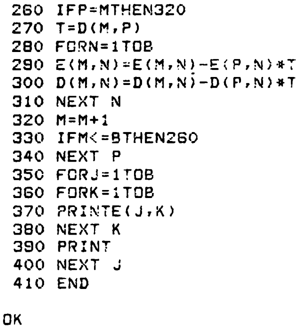


```
 280 PRINT
 290 FORJ=1TO3
 300 FORK=1TO3
 310 I=I+1
 320 Y(J,K)=(D(I,1)*D(I,4)-D(I,2)*D(I,3))
 330 Y(J,K)=Y(J,K)*((-1)^(J+K))
 340 PRINTY(J,K)
 350 NEXT K
 360 PRINT
 370 NEXT J
 380 END

OK
```

Matrix Inversion

MATRIX INVERSION - In this program, the size of the matrix must be entered first; then the numbers of the matrix, by lines. The program then will give the inverse matrix , also known as A to the -1, of the original matrix.

```
10 REM MATRIX INVERSION
20 FORG=1TO30:PRINT:NEXT
30 INPUT"RANK OF MATRIX";B
40 PRINT
50 FORJ=1TOB
60 FORK=1TOB
70 INPUTD(J,K)
80 NEXT K
90 PRINT
100 NEXT J
110 FORJ=1TOB
120 FORK=1TOB
130 E(J,K)=0
140 IFJ<>KTHEN160
150 E(J,K)=1
160 NEXT K
170 NEXT J
180 FORP=1TOB
190 T=D(P,P)
200 IFT=0THEN410
210 FORN=1TOB
220 E(P,N)=(E(P,N))/T
230 D(P,N)=(D(P,N))/T
240 NEXT N
250 M=1
```



```
 260 IFP=MTHEN320
 270 T=D(M,P)
 280 FORN=1TOB
 290 E(M,N)=E(M,N)-E(P,N)*T
 300 D(M,N)=D(M,N)-D(P,N)*T
 310 NEXT N
 320 M=M+1
 330 IFM<=BTHEN260
 340 NEXT P
 350 FORJ=1TOB
 360 FORK=1TOB
 370 PRINTE(J,K)
 380 NEXT K
 390 PRINT
 400 NEXT J
 410 END

OK
```

Peculiar Value of 3/3 Matrix

PECULIAR VALUE OF 3/3 MATRIX - In this program, the 3 x 3 matrix is entered. The program then calculates the peculiar value of this matrix.

```
 10 REM PECULIAR VALUE OF 3/3 MATRIX
 20 FORQ=1TO30:PRINT:NEXT
 30 FORJ=1TO3
 40 PRINT
 50 FORK=1TO3
 60 INPUTY(J,K)
 70 NEXT K
 80 NEXT J
 90 FORJ=1TO3
 100 X(J,J)=Y(J,J)
 110 NEXT J
 120 PRINT
 130 FORL=-10TO20
 140 FORJ=1TO3
 150 Y(J,J)=Y(J,J)-L
 160 NEXT J
 170 P2=-1*Y(1,2)*(Y(2,1)*Y(3,3)-Y(2,3)*Y(3
,1))
 180 P3=Y(1,3)*(Y(2,1)*Y(3,2)-Y(2,2)*Y(3,1)
)
 190 P=P1+P2+P3
 200 FORJ=1TO3
 210 Y(J,J)=X(J,J)
 220 NEXT J
 230 PRINT"L";L,"P(L)";P
 240 NEXT L
 250 END
```

OK

System of Linear Equations

SYSTEM OF LINEAR EQUATIONS - If you have to solve a system of N linear equations, then use this program. First, enter N, then enter the co-efficients of the first equation, then the second, and so on. The last entry will be the solution vector. The program then will give the values of all of the unknowns.

```
10 REM SYSTEM OF LINEAR EQUATIONS
20 REM ENTER COEFFICIENTS BY LINES
30 FORQ=1TO30:PRINT:NEXT
40 INPUT"RANK OF THE SYSTEM";B
50 PRINT
60 PRINT"MATRIX COEFFICIENTS"
70 FORJ=1TOB
80 FORK=1TOB
90 INPUTD(J,K)
100 NEXT K
110 PRINT
120 NEXT J
130 GOSUB310
140 PRINT"INVERSE C.MAT."
150 FORJ=1TOB
160 FORK=1TOB
170 PRINTE(J,K)
180 NEXTK
```

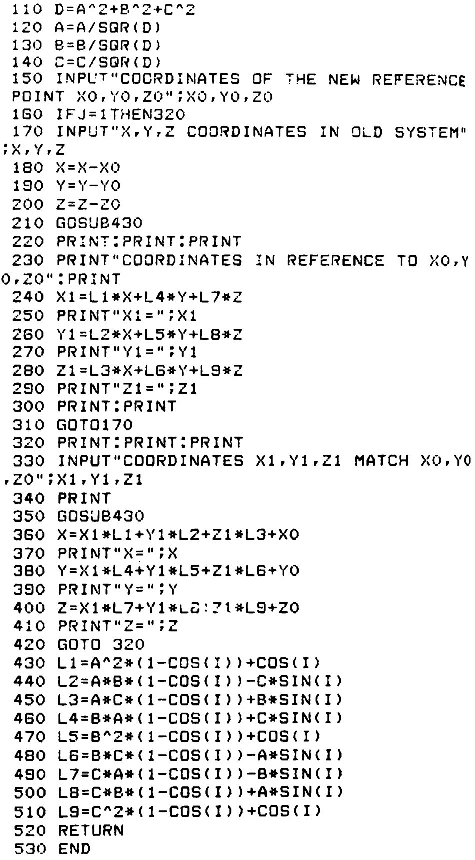
```
190 PRINT
200 NEXT J
210 PRINT"COLUMN OF THE EXPANDED MATRIX"
220 C=1
230 FORJ=1TOB
240 FORK=1TOC
250 INPUTY(J,K)
260 NEXT K
270 PRINT
280 NEXT J
290 GOSUB570
300 STOP
310 FORJ=1TOB
320 FORK=1TOB
330 E(J,K)=0
340 IFJ<>KTHEN360
350 E(J,K)=1
360 NEXT K
370 NEXT J
380 FORP=1TOB
390 T=D(P,P)
400 IFT=0THEN560
410 FORN=1TOB
420 E(P,N)=E(P,N)/T
430 D(P,N)=D(P,N)/T
440 NEXT N
450 M=1
460 IFP=MTHEN520
470 T=D(M,P)
480 FORN=1TOB
490 E(M,N)=E(M,N)-E(P,N)*T
500 D(M,N)=D(M,N)-D(P,N)*T
510 NEXT N
520 M=M+1
530 IFM<=BTHEN460
540 NEXT P
550 RETURN
560 STOP
570 S=0
580 FORI=1TOB
590 FORK=1TOC
600 FORJ=1TOB
```

```
610 S=E(I,J)*Y(J,K)+S
620 NEXT J
630 Z(I,K)=S
640 S=0
650 NEXT K
660 NEXT I
670 PRINT
680 PRINT
690 PRINT"SOLUTION-VECTOR:"
700 PRINT:PRINT
710 FORI=1TOB
720 FORK=1TOC
730 PRINTZ(I,K)
740 NEXT K
750 PRINT
760 NEXT I
770 RETURN
```

Co-ordinant Transformation

CO-ORDINANT TRANSFORMATION - This program gives the new co-ordinants after translation and rotation of the system.

```
 10 REM COÖRDINATE TRANSFORMATION BY TRANSL
ATING AND ROTATING
 20 FORQ=1TO30:PRINT:NEXT
 30 INPUT"DO YOU WANT TRANSFORMATION FROM S
YSTEM 0,0,0 TO X0,Y0,Z0";K$
 40 J=0
 50 IFK$="Y"THEN70
 60 J=1
 70 PRINT
 80 INPUT"ROTATION ANGLE IN DEGREES";I
 90 I=6.283185*(I/360)
 100 INPUT"X,Y,Z COORDINATES OF ROTATION-AX
IS";A,B,C
```



```
 110 D=A^2+B^2+C^2
 120 A=A/SQR(D)
 130 B=B/SQR(D)
 140 C=C/SQR(D)
 150 INPUT"COORDINATES OF THE NEW REFERENCE
 POINT X0,Y0,Z0";X0,Y0,Z0
 160 IFJ=1THEN320
 170 INPUT"X,Y,Z COORDINATES IN OLD SYSTEM"
;X,Y,Z
 180 X=X-X0
 190 Y=Y-Y0
 200 Z=Z-Z0
 210 GOSUB430
 220 PRINT:PRINT:PRINT
 230 PRINT"COORDINATES IN REFERENCE TO X0,Y
0,Z0":PRINT
 240 X1=L1*X+L4*Y+L7*Z
 250 PRINT"X1=";X1
 260 Y1=L2*X+L5*Y+L8*Z
 270 PRINT"Y1=";Y1
 280 Z1=L3*X+L6*Y+L9*Z
 290 PRINT"Z1=";Z1
 300 PRINT:PRINT
 310 GOTO170
 320 PRINT:PRINT:PRINT
 330 INPUT"COORDINATES X1,Y1,Z1 MATCH X0,Y0
,Z0";X1,Y1,Z1
 340 PRINT
 350 GOSUB430
 360 X=X1*L1+Y1*L2+Z1*L3+X0
 370 PRINT"X=";X
 380 Y=X1*L4+Y1*L5+Z1*L6+Y0
 390 PRINT"Y=";Y
 400 Z=X1*L7+Y1*L8:Z1*L9+Z0
 410 PRINT"Z=";Z
 420 GOTO 320
 430 L1=A^2*(1-COS(I))+COS(I)
 440 L2=A*B*(1-COS(I))-C*SIN(I)
 450 L3=A*C*(1-COS(I))+B*SIN(I)
 460 L4=B*A*(1-COS(I))+C*SIN(I)
 470 L5=B^2*(1-COS(I))+COS(I)
 480 L6=B*C*(1-COS(I))-A*SIN(I)
 490 L7=C*A*(1-COS(I))-B*SIN(I)
 500 L8=C*B*(1-COS(I))+A*SIN(I)
 510 L9=C^2*(1-COS(I))+COS(I)
 520 RETURN
 530 END
```

Geometry

GEOMETRY - This program is useful if you have to work with planes in three dimensions.

```
 10 REM---GEOMETRY---
 20 FORQ=1TO30:PRINT:NEXT
 30 PRINT"YOU KNOW :"
 40 PRINT:PRINT"-1--NORMALIZED EQUATION OF
THE PLANE"
 50 PRINT:PRINT"-2--POINT OF INTERSECTION O
F LINE AND PLANE"
 60 PRINT:PRINT"-3--DISTANCE POINT TO PLANE
":PRINT
 70 PRINT"-4--LINE OF INTERSECTION BETWEEN
2 PLANES":PRINT
 80 PRINT"-5--ANGLE BETWEEN 2 LINES":PRINT
 90 PRINT
 100 INPUT"ENTER ONE NUMBER";R
 110 PRINT:PRINT:PRINT:PRINT:PRINT
 120 ONRGOTO130,140,150,160,170
 130 GOSUB1080:STOP
 140 GOSUB400:STOP
 150 GOSUB580:STOP
 160 GOSUB670:STOP
 170 GOSUB180:STOP
 180 PRINT"ANGLE BETWEEN TWO LINES"
 190 PRINT"VECTORS OF THE LINES"
 200 PRINT"LINE 1"
 210 INPUTX1,X2,X3
 220 PRINT"LINE 2
 230 PRINT
 240 INPUTG1,G2,G3
 250 PRINT
 260 PRINT
 270 C=X1*G1+X2*G2+X3*G3
 280 D=SQR(X1^2+X2^2+X3^2)
 290 E=SQR(G1^2+G2^2+G3^2)
 300 C=C/(D*E)
 310 PRINT"COS A="C
 320 C=C/SQR(1-C^2)
 330 C=1.570796-ATN(C)
 340 PRINT:PRINT
 350 PRINT"ANGLE";C"RAD"
 360 C=C/3.1459275
 370 C=C*180
 380 PRINT"IN DEG :";C
 390 RETURN
 400 PRINT"POINT OF INTERSECTION(LINE/PLANE
)"
 410 T=0
 420 PRINT
 430 PRINT"POINT ON THE LINE"
 440 INPUTP1,P2,P3
 450 PRINT
 460 PRINT"VECTOR OF THE LINE"
 470 INPUTG1,G2,G3
 480 GOSUB1080
 490 T=(C-NX*P1-NY*P2-NZ*P3)/(G1*NX+G2*NY+G
3*NZ)
 500 X=P1+G1*T
 510 Y=P2+G2*T
 520 Z=P3+G3*T
 530 PRINT
 540 PRINT"POINT OF INTERSECTION"
 550 PRINT
 560 PRINT"X";X;"Y";Y;"Z";Z
 570 STOP
 580 PRINT"DISTANCE(POINT/PLANE)
 590 PRINT
 600 PRINT"COORDINATES OF THE POINT"
 610 INPUTP1,P2,P3
 620 GOSUB1080
 630 D=NX*P1+NY*P2+NZ*P3-C
 640 PRINT"DISTANCE=";D
 650 PRINT
 660 RETURN
 670 REM LINE OF INTERSECTION
 680 PRINT
 690 PRINT"PLANE-1-"
 700 GCSUB1080
 710 EX=NX
 720 EY=NY
 730 EZ=NZ
 740 CE=C
 750 PRINT
 760 PRINT
 770 PRINT"PLANE-2-"
 780 PRINT
 790 GOSUB1080
 800 AX=EX
 810 AY=EY
 820 AZ=EZ
 830 BX=NX
 840 BY=NY
 850 FX=NX
 860 FY=NY
 870 FZ=NZ
 880 GOSUB1700
 890 PRINT:PRINT"VECTOR OF THE LINE OF INTE
RSECTION"
 900 PRINT"X";NX;"Y";NY;"Z";NZ
 910 PRINT
 920 IF(FY*EZ-FZ*EY)=0THEN970
 930 PRINT
 940 Z=(FY*CE-EY*C)/(FY*EZ-FZ*EY)
 950 PRINT"POINT ON THE LINE OF INTERSECTIO
N"
 960 PRINT0,Y,Z
 970 IF(FX*EY-FY*EX)=0THEN1030
 980 X=(C*EY-FY*CE)/(FX*EY-FY*EX)
 990 Y=(FX*CE-C*EX)/(FX*EY-FY*EX)
 1000 PRINT"POINT ON THE LINE OF INTERSECTI
ON"
 1010 PRINT"X";X;"Y";Y;"Z";Z
 1020 RETURN
 1030 X=(C*EZ-FZ*CE)/(FX*EZ-FZ*EX)
 1040 Z=(FX*CE-C*EX)/(FX*EZ-FZ*EX)
 1050 PRINT"POINT ON THE LINE OF INTERSECTI
ON"
```

Number Converter

NUMBER CONVERTER - First the Base is entered; then, the number. The program will convert to decimal value. Base 2 to Base 16 can be used.

```
10 REM NUMBER CONVERTER
20 FORQ=1TO30:PRINT:NEXT
30 INPUT"ENTER BASE";B1
40 IF B1<2 OR B1>16 THEN 30
50 INPUT"ENTER NUMBER";N1$
60 L=LEN(N1$)
70 DEC=0
80 P=0
90 FOR J=L TO 1 STEP -1
100 K=ASC(MID$(N1$,J,1))
110 IF K>64 THEN K=K-7
120 K=K-48
130 IF K<B1 AND K>-1 THEN 170
140 PRINT:PRINT"INVALID INPUT FOR BASE";B1
150 PRINT:PRINT:PRINT
160 GOTO 30
170 DEC=DEC+K*B1^P
180 P=P+1
190 NEXT J
200 PRINT:PRINT"DECIMAL VALUE=";DEC
210 PRINT:PRINT:PRINT
220 GOTO 30
230 END

OK
```

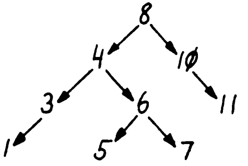
Sorting (Binary Tree)

SORTING ALGORITHM (BINARY TREE) - Sometimes it is necessary to sort data in memory. If there is a lot of data, it is important to have a fast sorting algorithm. But since the fast algorithms need more memory, you have to compromise.

This algrithm uses the method of the binary tree. The data will be stored in the form of a table. This data will be sorted but there are pointer-arrays that define where the next smaller and next larger elements are in the table.

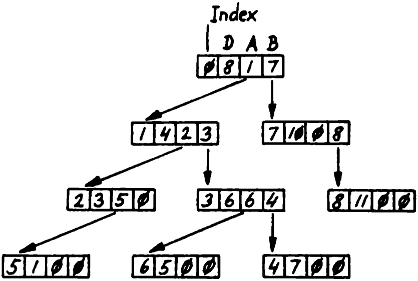
SAMPLE:

The following numbers are to be sorted:
8,4,3,6,7,1,5,10,11
The binary tree looks like this:
(<=corresponds to left edge, > corresponds to right edge)



OUTPUT OF THE SORTED TREE - To get the sorted sequence you have to run through the tree from bottom left to bottom right. A left edge has precedence over a junction and a junction over the right edge.

Action of The Tree in BASIC:



	No.	<>	>
Index	D	A	B
0	8	1	7
1	4	2	3
2	3	5	-
3	6	6	4
4	7	-	-

5	*1*	-	-
6	*5*	-	-
7	*10*	-	*8*
8	*11*	-	-

```
 10 REM SORTING
 20 FORQ=1TO30:PRINT:NEXT
 30 INPUT"NUMBER OF ELEMENTS";N
 40 PRINT:PRINT
 50 N=N-1
 60 K=N
 70 DIM A(N),B(N),C(N),D(N)
 80 FORI=0TON:D(I)=INT(100*RND(N)+.5)
 90 PRINTD(I):NEXT:PRINT
 100 REM *** SORTING ***
 110 FORI=0TOK-1
 120 L=0
 130 IFD(I+1)>D(L)GOTO160
 140 IFA(L)=0THENA(L)=I+1:GOTO180
 150 L=A(L):GOTO130
 160 IFB(L)=0THENB(L)=I+1:GOTO180
 170 L=B(L):GOTO130
 180 NEXTI
 190 REM *** OUTPUT ***
 200 M=-1:L=0
 210 I=0
 220 IFA(L)>0THENM=M+1:C(M)=L:L=A(L):GOTO22
0
 230 IFI>KGOTO340
 240 PRINTD(L)
 250 I=I+1
 260 IFB(L)=0GOTO280
 270 L=B(L):GOTO220
 280 IFM<0GOTO340
 290 I=I+1:PRINTD(C(M)):L=B(C(M))
 300 IFI>KGOTO340
 310 M=M-1
 320 IFL=0GOTO280
 330 GOTO220
 340 END
```

Numerical Differentiation

NUMERICAL DIFFERENTIATION - This program gives the value of the first and second derivative at position X of the in line 50 defined equation.

```
10 REM NUMERICAL DIFFERENTIATION
20 REM FIRST AND SECOND DERIVITIVES
30 FORQ=1TO30:PRINT:NEXT
40 INPUT"X=";X
50 DEFFNY(X)=X^2+2*X
60 D=.01
70 A(1)=(FNY(X+.5*D)-FNY(X-.5*D))/D
80 A(2)=(FNY(X)-FNY(X-D))/D
90 A(3)=(FNY(X+D)-FNY(X))/D
100 A(4)=(A(3)-A(2))/D
110 PRINT:PRINT:PRINT"F(X)=";FNY(X)
120 PRINT:PRINT"F1(X)=";A(1)
130 PRINT:PRINT"F2(X)=";A(4)
140 END

OK
```

Numerical Integration (Simpson)

NUMERICAL INTEGRATION - This program calculates (by the Simpson procedure), the value of the integration of the in line 30 defined equation between the points A and B.

```
 10 REM NUMERICAL INTEGRATION (SIMPSON)
 20 FORQ=1TO30:PRINT:NEXT
 30 DEFFNY(X)=X^3
 40 INPUT"BEGINNING";A
 50 PRINT:INPUT"END";B
 60 N=100
 70 D=(B-A)/N
 80 S=FNY(B)+FNY(A)
 90 FORJ=0TON-2STEP2
 100 S=S+4*FNY(A+(J+1)*D)+2*FNY(J*D+A)
 110 NEXT J
 120 S=S*D/3
 130 PRINT:PRINT:PRINT
 140 PRINT"S=";S
 150 END

OK
```

Differential Equation

DIFFERENTIAL EQUATION - This program needs 4300 bytes of memory. It contains a routine for CLEAR SCREEN.

The Lotha-Volterra-model describes the evolution of two types of animals, dependent upon each other (PREY-CARNIVORE). This is a model that shows the area of the differential equations.

There are two differential equations:

Y'(t)=-ALPHA(Y(t))+BETA(Y(t))(X(t))
X'(t)=GAMMA(X(t))-DELTA(X(t))(Y(t))

with the starting values:

Y(0)=Y0 and X(0)=X0 where

Y(t) is the evolution of the carnivores and
X(t) is the evolution of the prey

If BETA and DELTA = 0, then the breeds will evolve independently. Y(t) will decline exponentially, and X(t) will grow exponentially.

The RUNGE-KUTTA procedure is a way to solve a differential equation numerically.

Differential Equation (continued)

Concerning the CURSOR:

If the CURSOR appears (character 161), you can continue by hitting RETURN. After the graphics are finished, the CURSOR appears. You can hit A or Z. With A you jump to the location where the co-ordinates (MIN,MAX) may be changed; with Z, to the location for changing the parameters of the model.

If a change is desired, move the CURSOR up(A=up) or down(Z=down), to the line you want to change, then hit RETURN. The selected line appears as an input line and you can change the parameters of the line.

The meaning of this model:

For many years man has used mathematics to describe proceedings in nature. Physical values are described by mathematical relations. With "mathematical systems" you can describe more complex proceedings. If you have found a model that describes the proceedings well enough, then you can help threatened species (whales) or improve pest control. This example will be a demonstration for mathematical models of nature.

```
 10 REM DEQ
 20 RESTORE
 30 POKE11,68:POKE12,2
 40 FORT=580TO603:READE:POKET,E:NEXT
 50 DATA169,32,160,4,162,0,157,0,208,232,20
8
 60 DATA250,238,76,2,136,208,244,169,208,14
1,76,2,96
 70 GOSUB830
 80 PRINT" LOTKA-VOLTERRA MODEL":PRINT
 90 PRINT"  --> (RETURN)";
 100 GOSUB960:PRINT
 110 DATA20,20,6,4,.2,.2,.04,50
 120 READB:READA:READAL:READBE:READGA:READD
E:READH:READNN
 130 XA=NN:MA=NN
 140 GOSUB830
 150 PRINT"  COORDINATES:":PRINT:PRINT
 160 PRINT"1 XMINIMUM=";XI:PRINT
 170 PRINT"2 XMAXIMUM=";XA:PRINT
 180 PRINT"3 YMINIMUM=";MI:PRINT
 190 PRINT"4 YMAXIMUM=";MA:PRINT
 200 PRINT:PRINT" CHANGE COORDINATES ?"
 210 GOSUB960
 220 IFZ=47ORZ=-1GOTO300
 230 PRINT:PRINT
 240 ONZGOTO260,270,280,290
 250 GOTO210
 260 INPUT"  XMINIMUM=";XI:GOTO140
 270 INPUT"  XMAXIMUM=";XA:GOTO140
 280 INPUT"  YMINIMUM=";MI:GOTO140
 290 INPUT"  YMAXIMUM=";MA:GOTO140
 300 GOSUB830
 310 PRINT"  GRAFICS:":PRINT
 320 PRINT
 330 PRINT"1 CARNIVORE-QUARRY":PRINT
 340 PRINT"2 TIME-CARNIVORE":PRINT
 350 PRINT"3 TIME-QUARRY":PRINT
 360 GOSUB960
 370 IFZ=-1ORZ=47THENZ=1
 380 Z1$="TIME":Z2$="QUARRY"
 390 IFZ=1THENZ1$="CARNIVORE"
 400 IFZ=2THENZ2$="CARNIVORE"
 410 AZ=Z
 420 GOSUB830:GOSUB1060
 430 POKE54117,161:KE=57088:POKE530,1:POKEK
E,253
 440 IFPEEK(KE)=223THENPOKE530,0:GOTO420
 450 IFPEEK(KE)=191THENPOKE530,0:GOTO140
 460 GOTO440
 470 REM GRAFIC3 UP
 480 REM XI,XA,MI,MA .
 490 IFL<MIORL>MAORJ<XIORJ>XATHENRETURN
 500 I=INT((J-XI)/(XA-XI)*43)
 510 K=INT((L-MI)/(MA-MI)*45)
 520 UA=INT(I/2)+54055
 530 MX=I-INT(I/2)*2
 540 MY=K-INT(K/2)*2
 550 AD=UA-32*(K-MY)/2
 560 PE=PEEK(AD)
 570 IFPE=169ORP=170OR(PE>153ANDPE<158)THEN
RETURN
 580 IFPE<>32ANDNOT(PE=128ORPE=209ORPE=136)
GOTO600
 590 POKEAD,167+MY-2*MX:RETURN
 600 IFPE<>165GOTO650
 610 IFMX=0ANDMY=0THENZZ=154:GOTO810
 620 IFMX=0ANDMY=1THENZZ=169:GOTO810
 630 IFMX=1ANDMY=0THENZZ=165:GOTO810
 640 ZZ=157:GOTO810
 650 IFPE<>166GOTO710
 660 IFMX=0ANDMY=0THENZZ=170:GOTO810
 670 IFMX=0ANDMY=1THENZZ=155:GOTO810
 680 IFZ+AB=4ORZ+AB=6THENZZ=153+Z:GOTO810
 690 IFMX=1ANDMY=0THENZZ=157:GOTO810
 700 ZZ=166:GOTO810
 710 IFPE<>167GOTO760
 720 IFMX=0ANDMY=0THENZZ=167:GOTO810
 730 IFMX=8ANDMY=1THENZZ=156:GOTO810
 740 IFMX=1ANDMY=0THENZZ=154:GOTO810
 750 ZZ=170:GOTO810
 760 IFPE<>168GOTO820
 770 IFMX=0ANDMY=0THENZZ=156:GOTO810
 780 IFMX=0ANDMY=1THENZZ=168:GOTO810
 790 IFMX=1ANDMY=0THENZZ=169:GOTO810
 800 ZZ=155
 810 POKEAD,ZZ
 820 RETURN
 830 WW=USR(WW)
 840 RETURN
 850 REM LETTERING
 860 Z1=53351:Z2=54055:Z3=54076
 870 FORT=Z1TOZ2STEP32:POKET,136:NEXTT
 880 FORT=Z2TOZ3:POKET,128:NEXTT:POKEZ2,209
 890 Z4=32:Z$=MID$(STR$(MA),2):Z5=53318:GOS
UB1040
```

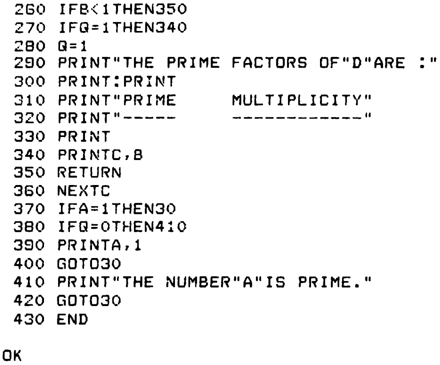
```
 900 Z4=32:Z$=MID$(STR$(MI),2):Z5=54054-LEN
(Z$)*Z4:GOSUB1040
 910 Z4=1:Z$=MID$(STR$(XI),2):Z5=54086:GOSU
B1040
 920 Z4=1:Z$=MID$(STR$(XA),2):Z5=54108-LEN(
Z$):GOSUB1040
 930 Z4=32:Z$=Z2$:Z5=53445:GOSUB1040
 940 Z4=1:Z$=Z1$:Z5=54124:GOSUB1040
 950 RETURN
 960 KE=57088:POKE530,1
 970 POKEKE,223:R=54117
 980 Z=PEEK(R):POKER,161:QQ=1^1^1
 990 POKER,161
 1000 IFPEEK(KE)=191THENPOKER,Z:R=R-32:GOTO
980
 1010 IFPEEK(KE)=223THENPOKER,Z:R=R+32:GOTO
980
 1020 POKEKE,223:IFPEEK(KE)<>247THENPOKEKE,
253:GOTO1000
 1030 POKE530,0:Z=Z-48:RETURN
 1040 FORT=Z4TOLEN(Z$)*Z4STEPZ4:POKEZ5+T,AS
C(MID$(Z$,T/Z4,1))
 1050 NEXTT:RETURN
 1060 GOSUB830:PRINT"1 CARN.STARTING V.=";B
 1070 PRINT:PRINT"2 QUA.STARTING V.=";A
 1080 PRINT:PRINT"3 DEATH-RATE CARN.=";ALPH
A
 1090 PRINT:PRINT"4 BIRTH-RATE QUA.=";BETA
 1100 PRINT:PRINT"5 CONS.-RATE CARN.=";GAMM
A
 1110 PRINT:PRINT"6 HIT-RATE QUA.=";DELTA
 1120 PRINT:PRINT"7 STEPWIDTH=";H
 1130 PRINT:PRINT"8 STEPS N=";NN
 1140 PRINT:PRINT:PRINT"  CHANGE DATES ?":G
OSUB960
 1150 II=Z:IFZ=47ORZ=-16GOTO1330
 1160 ONIIGOTO1170,1190,1210,1230,1250,1270
,1290,1310
 1170 PRINT:INPUT" CARN. STARTING VALUE=";B
 1180 PRINT:GOTO1060
 1190 PRINT:INPUT" QUA. STARTING VALUE=";A
 1200 PRINT:GOTO1060
 1210 PRINT:INPUT" DEATH-RATE CARN.=";AL
 1220 PRINT:GOTO1060
```

```
1230 PRINT:INPUT" BIRTH-RATE QUA.=";BE
1240 PRINT:GOTO1060
1250 PRINT:INPUT" CONS.-RATE CARN.=";GA
1260 PRINT:GOTO1060
1270 PRINT:INPUT" HIT-RATE QUA.=";DE
1280 PRINT:GOTO1060
1290 PRINT:INPUT" STEP WIDTH=";H
1300 PRINT:GOTO1060
1310 PRINT:INPUT" STEPS N=";NN
1320 PRINT:GOTO1060
1330 GOSUB830:N=NN
1340 AA=A:BB=B
1350 GOSUB850
1360 FORT=0TON
1370 GOSUB1460
1380 GOSUB1580
1390 IFAZ=1THENJ=B:L=A:GOTO1420
1400 IFAZ=2THENJ=T:L=B:GOTO1420
1410 IFAZ=3THENJ=T:L=A
1420 GOSUB470
1430 NEXTT
1440 A=AA:B=BB
1450 RETURN
1460 REM RUNGE-KUTTA
1470 REM PAR UJ,H,B
1480 UJ=B:I=1:GOSUB1660
1490 UJ=B+V(1)/2:I=2:GOSUB1660
1500 UJ=B+V(2)/2:I=3:GOSUB1660
1510 UJ=B+V(3):I=4:GOSUB1660
1520 A(1)=1/6:A(2)=1/3:A(3)=1/3:A(4)=1/6
1530 FORI=1TO4
1540 B=B+A(I)*V(I)
1550 NEXTI
1560 RETURN
1570 REM PAR UI,H,A
1580 UI=A:I=1:GOSUB1670
1590 UI=A+V(1)/2:I=2:GOSUB1670
1600 UI=A+V(2)/2:I=3:GOSUB1670
1610 UI=A+V(3):I=4:GOSUB1670
1620 FORI=1TO4
1630 A=A+A(I)*V(I)
1640 NEXTI
1650 RETURN
1660 V(I)=H*(-AL*UJ+GA*A*UJ):RETURN
1670 V(I)=H*(BE*UI-DE*B*UI):RETURN
```

Prime Factors

PRIME FACTORS - This program solves for prime numbers and gives the number and its power to equal a designated number. The program can also be used to determine if a number is a prime number. It is accurate to more than 134,000,000.

```
10 REM PRIME FACTORS
20 FORQ=1TO25:PRINT:NEXT
30 PRINT:PRINT:PRINT:PRINT:PRINT
40 INPUT"NUMBER";A
50 IFA<134217728THEN90
60 PRINT"SORRY! TOO BIG"
70 PRINT
80 GOTO30
90 D=A
100 FORQ=1TO30:PRINT:NEXT
110 IFA=2THEN410
120 Q=0
130 IFA>0THEN150
140 STOP
150 C=2
160 GOSUB200
170 FORC=3TOSQR(A)STEP2
180 GOSUB200
190 GOTO360
200 B=0
210 IFA=C*INT(A/C)THEN230
220 GOTO260
230 A=A/C
240 B=B+1
250 GOTO210
```



```
 260 IFB<1THEN350
 270 IFQ=1THEN340
 280 Q=1
 290 PRINT"THE PRIME FACTORS OF"D"ARE :"
 300 PRINT:PRINT
 310 PRINT"PRIME     MULTIPLICITY"
 320 PRINT"-----     ------------"
 330 PRINT
 340 PRINTC,B
 350 RETURN
 360 NEXTC
 370 IFA=1THEN30
 380 IFQ=0THEN410
 390 PRINTA,1
 400 GOTO30
 410 PRINT"THE NUMBER"A"IS PRIME."
 420 GOTO30
 430 END

OK
```

Pythagorean Numbers

*PYTHAGORIAN NUMBERS - This program searches for all possible variables in the quadratic equation A*A+B*B=C*C. The operator types in the amount he or she wants "C" to equal.*

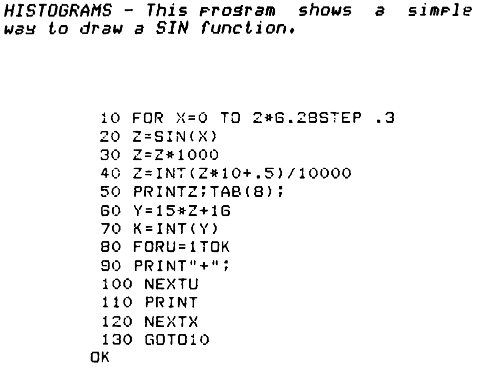
```
 10 REM PYTHAGOREAN NUMBERS
 20 FORX=1TO30:PRINT:NEXT
 30 DEFFNY(X)=INT(X*10+.5)/10
 40 INPUT"UP TO WHICH NUMBER";G
 50 FORX=1TO30:PRINT:NEXT
 60 PRINT"  A*A + B*B = C*C"
 70 K=1:PRINT
 80 FORL=0TOK-1:M=2*K:N=2*L+1
 90 FORX=2TON+1:Y=M/N:IFY<>INT(Y)THEN120
 100 Z=N/X:IFZ<>INT(Z)THEN120
 110 GOTO170
 120 NEXTX:N=2*L+1:A=M*M-N*N:B=2*M*N
 130 C=M*M+N*N:D=A*A+B*B:E=C*C
 140 A=FNY(A):B=FNY(B):C=FNY(C)
 150 IFC>GTHEN260
 160 GOSUB270
 170 M=2*K+1:N=2*L+2
 180 FORX=2TON:Y=M/X:IFY<>INT(Y)THEN210
 190 Z=N/X:IFZ<>INT(Z)THEN210
 200 GOTO250
 210 NEXTX:A=M*M-N*N:B=2*M*N:C=M*M+N*N:D=A*
A+B*B:E=C*C
 220 A=FNY(A):B=FNY(B):C=FNY(C)
 230 IFC>GTHEN260
 240 GOSUB270
 250 NEXTL:K=K+1:GOTO80
 260 END
 270 PRINTTAB(2);A;TAB(8);B;TAB(14);C
 280 RETURN
```

Decibel Program

DECIBEL PROGRAM - This program calculates the change of either the power or voltage ratio in decibels in electrical circuits.

```
 10 REM DECIBEL PROGRAM
 20 FORX=1TO25:PRINT:NEXT
 30 PRINT"DECIBELS FOR VOLTAGE AND POWER RA
TIOS"
 40 PRINT:INPUT" ENTER MODE (V OR P) ";X$
 50 IFX$="V"THEN110
 60 I=10
 70 PRINT:INPUT" ENTER P1 IN WATTS ";P1
 80 PRINT:INPUT" ENTER P2 IN WATTS ";P2
 90 A=P1:B=P2
 100 GOTO150
 110 PRINT:INPUT" ENTER E1 IN VOLTS ";E1
 120 PRINT:INPUT" ENTER E2 IN VOLTS ";E2
 130 A=E1:B=E2:I=20
 140 B=E2:I=20
 150 X=I * LOG (A/B)/LOG (10)
 160 X=X*1000:PRINT:PRINT
 170 X=INT(X*10+.5)/10000
 180 PRINT"THE "X$;" RATIO REPRESENTS A CHA
NGE OF ";X;" DECIBELS"
 190 PRINT:PRINT:PRINT:PRINT:GOTO40
OK
```

Histograms



HISTOGRAMS - This program shows a simple way to draw a SIN function.

```
 10 FOR X=0 TO 2*6.28STEP .3
 20 Z=SIN(X)
 30 Z=Z*1000
 40 Z=INT(Z*10+.5)/10000
 50 PRINTZ;TAB(8);
 60 Y=15*Z+16
 70 K=INT(Y)
 80 FORU=1TOK
 90 PRINT"+";
 100 NEXTU
 110 PRINT
 120 NEXTX
 130 GOTO10
OK
```

Regression Analysis

REGRESSION ANALYSIS - The operator enters at least three pairs of numbers from a carthesian coordinate system. The program then lists the results of regression analysis.

```
 10 REM REGRESSION ANALYSIS
 20 FORQ=1TO30:PRINT:NEXT
 30 DEFFNY(X)=INT(X*1000+.5)/1000
 40 DIML(2,50)
 50 INPUT"ENTER NUMBER OF PAIRS OF NUMBERS"
;N1:IFN1>=3THEN70
 60 PRINT"N HAS TO BE GREATER THAN 2":GOTO5
0
 70 PRINT"ENTER; 'X,Y'"
 80 FORN=1TON1:INPUTL(1,N),L(2,N):NEXT
 90 FORN=1TON1
 100 X=X+L(1,N)
 110 Y=Y+L(2,N)
 120 X2=X2+L(1,N)^2
 130 Y2=Y2+L(2,N)^2
 140 Z=Z+L(1,N)*L(2,N):NEXT
 150 X1=X/N:Y1=Y/N
 160 B=(N*Z-X*Y)/(N*X2-X^2):A=Y1-B*X1
 170 S1=((Y2-A*Y-B*Z)/(N-2))^.5
 180 X0=(N*X2-X*X)^.5
 190 Y0=(N*Y2-Y*Y)^.5
 200 R=(N*Z-X*Y)/(X0*Y0)
 210 Z=FNY(Z):X2=FNY(X2):Y2=FNY(Y2)
 220 X1=FNY(X1):Y1=FNY(Y1):B=FNY(B)
 230 A=FNY(A):S1=FNY(S1):R=FNY(R)
 240 FORQ=1TO30:PRINT:NEXT
 250 PRINT"RESULTS OF REGRESSION ANALYSIS"
 260 PRINT
 270 PRINT"NUMBER OF PAIRS";N1
 280 PRINT"SUM OF X- VALUES";X
 290 PRINT"SUM OF Y-VALUES";Y:PRINT
 300 PRINT"SUM OF X*Y";Z
 310 PRINT"SUM OF X-SQUARES";X2
 320 PRINT"SUM OF Y-SQUARES"Y2:PRINT
 330 PRINT"MEDIAN OF X";X1
 340 PRINT"MEDIAN OF Y";Y1:PRINT
 350 PRINT"SLOPE B OF STRAIGHT LINE";B
 360 PRINT"POINT OF INTERSECTION WITH Y-AXI
S";A
 370 PRINT
 380 PRINT"STANDARD FAULT";S1
 390 PRINT"CORRELATION-COEFFICIENT";R
 400 END

OK
```

Simple Statistics

SIMPLE STATISTICS - The operator enters starting values. For each value of measurement the repeat factor is entered. The result will show the mean, median, and the variation coefficient.

```
 10 REM SIMPLE STATISTICS
 20 FORQ=1TO30:PRINT:NEXT:CLEAR
 30 PRINT "SIMPLE STATISTICS" : PRINT
 40 PRINT "BEGINNING VALUE:"
 50 INPUT"NUMBER (N)";N
 60 INPUT"SUM OF X (SX)";SX
 70 INPUT"SUM OF X-SQUARE (QX)";QX
 80 PRINT:PRINT:PRINT
 90 PRINT "VALUES OF MEASUREMENT:"
 100 PRINT
 110 INPUT "VALUE OF MEASUREMENT, NUMBER";
X, I
 120 IFI=0THEN170
 130 N = N + I
 140 SX = SX + X * I
 150 QX = QX + X * X * I
 160 GOTO110
 170 PRINT:PRINT:PRINT:PRINT:PRINT
 180 M=SX/N
 190 S = SQR((QX - M * SX)/(N-1))
 200 V = S * 100 / M
 210 PRINT
 220 PRINT"NUMBER (N)";N
 230 PRINT"SUM OF X (SX)";SX
 240 PRINT"SUM OF X-SQUARE (QX)";QX
 250 PRINT
 260 PRINT"MEDIAN (M)";M
 270 PRINT"STANDARD DEVIATION (S)";S
 280 PRINT"VARIATION COEFFICIENT (V)";V;"%"
 290 PRINT:PRINT:PRINT:PRINT:PRINT
 300 PRINT"ENTER :"
 310 PRINT
 320 PRINT "1 = EXTRA VALUES OF MEASUREMENT
"
 330 PRINT "2 = NEW ROW OF VALUES"
 340 INPUT I : PRINT
 350 ONIGOTO100,20
 360 GOTO300

OK
```

Function Plot

FUNCTION PLOT - This program plots your function (FNY(X)) defined in line 60 with three different periods (A,B,C). "L" is incremented after each loop. You can use "L" in line 60 so that you have a function (FNY(X,L)).

```
 10 REM FUNCTION PLOT
 20 FORQ=1TO30:PRINT:NEXT
 30 E=3:L=1
 40 INPUT"STARTING VALUE OF X";V
 50 T=3
 60 DEFFNY(X)=SIN(L*X)
 70 FORP=0TO2
 80 FORJ=53248TO54272:POKEJ,32:NEXT
 90 FORJ=53248TO54272STEP32:POKE(J+16),139
 100 NEXTJ
 110 POKE53392,120
 120 X=V
 130 FORJ=53248TO54203STEP32
 140 X=X+E*(P+1)
 150 U=FNY(X)
 160 POKE54088,(P+65)
 170 POKE54098,76:POKE54099,61
 180 POKE54100,(48+INT(L/10)):POKE54101,(48
+L-10*INT(L/10))
 190 C=INT(U)+16
 200 D=INT(8*(U-INT(U)))
 210 POKE(J+C),(136+D)
 220 IFABS(X)>(P*E)THEN240
 230 POKE(J+4),148:POKE(J+6),48:POKE(J+7),1
48:POKE(J+8),148
```

```
 240 NEXTJ
 250 NEXTP
 260 IFL=TTHEN20
 270 L=L+1
 280 P=0
 290 GOTO70

OK
```

Precipitation

PRECIPITATION - This program should be helpful in chemical analysis when the product and concentration of precipitation are required.

```
 10 REM PRECIPITAION
 20 FORQ=1TO30:PRINT:NEXT
 30 PRINT"OPTION MENU":PRINT
 40 PRINT"-1-PRODUCT OF SOLUABILITY":PRINT
 50 PRINT"-2-CONCENTRATION NECESSARY FOR PR
ECIPITATION":PRINT
 60 PRINT"-3-SOLUABILITY"
 70 PRINT
 80 INPUT"OPTION";U
 90 PRINT:PRINT:PRINT
 100 INPUT"SUBSTANCE 1 SEPARATES IN HOW MAN
Y NEGATIVE IONS ";K1
 110 PRINT:PRINT
 120 INPUT"IN HOW MANY ANIONS DOES SUBSTANC
E 1 SEPARATE ";A1
 130 ONUGOTO140,360,320
 140 PRINT:PRINT:PRINT
 150 INPUT"DO YOU KNOW THE SOLUBILITY OF SU
BSTANCE 1";Y$
 160 IFY$="N"THEN190
 170 INPUT"SOLUBILITY OF SUBSTANCE 1 IN MOL
/L";LM
 180 GOTO290
 190 INPUT"DO YOU KNOW THE CONCENTRATION OF
 THE NEGATIVE ION 1";Y$
 200 IFY$="N"THEN250
 210 INPUT" C";C1
 220 LP=C1^K1*(C1*A1/K1)^A1:PRINT
 230 PRINT:PRINT"PRODUCT OF SOLUBILITY=";LP
 240 GOTO310
 250 INPUT"SOLUBILITY OF THE ANION IN MOL/L
";C1
 260 LP=C1^A1*(C1*K1/A1)^K1
 270 PRINT:PRINT:PRINT
 280 PRINT"PRODUCT OF SOLUBILITY=";LP:GOTO3
10
 290 LP=(LM*K1)^K1*(LM*A1)^A1
 300 PRINT:PRINT"PRODUCT OF SOLUBILITY OF S
UBSTANCE 1=";LP:PRINT
 310 END
 320 INPUT"PRODUCT OF SOLUBILITY OF SUBSTAN
CE 1";LP
 330 C1=EXP((1/(K1+A1))*LOG(LP/(K1^K1*A1^A1
)))
 340 PRINT:PRINT:PRINT"SOLUBILITY=";C1;"MOL
/L"
 350 END
 360 PRINT:PRINT"PRODUCT OF SOLUBILITY OF T
HE" ;
 370 INPUT"CONCENTRATION TO PRECIPITATE";L
 380 PRINT:PRINT
 390 INPUT"ANION OR CATION (A/C)";I$
 400 PRINT
 410 IFI$="A"THEN480
 420 INPUT"CONCENTRATION OF THE CATION IN M
OL/L";C
 430 PRINT
 440 X=L/C^K1:X=1/A1*LOG(X):X=EXP(X)
 450 PRINT"TO PRECIPITATE THE CONCENTRATION
 THE ANION ";
 460 PRINT"HAS TO BE GREATER THAN";X;"MOL/L
"
 470 GOTO520
 480 PRINT"CONCENTRATION OF THE ANION IN MO
L/L";:INPUTC
 490 X=L/C^A1:X=1/K1*LOG(X):X=EXP(X)
 500 PRINT:PRINT"TO PRECIPITATE THE CONCENT
RATION THE CATION ";
 510 PRINT"HAS TO BE GREATER THAN";X;"MOL/L
"
 520 END
```

1.

IMPORTANT NEW BOOKS FROM ELCOMP

Order No.	Price	Title	Review
150	$11.00	Care and Feeding of The Commodore PET	Eight chapters exploring PET hardware. Includes repair and interfacing information Programming tricks and schematics.
151	$ 9.95	8K Microsoft BASIC Reference Manual	Authoritative reference manual for the original Microsoft 4K and 8K BASIC developed for Altair and later computers including PET. TRS-80, and OSI. OSI owners please take note!
152	$ 9.95	Expansion Handbook for 6502 and 6802 (S-44 Card Manual)	This is a revised, formal printing of our unique S-44 card manual. Describes all of the 4.5 x 6.5 44-pin S-44 cards including RAM, ROM, digital I/O, and MUX/A to D.
153	$14.90	Microcomputer Application Notes	Reprint of Intel literature
154	$ 6.95	Complex Sound Generation	New, revised applications manual for the Texas Instruments SN 76477 Complex Sound Generator. Circuit board available
155	$14.90	The First Book of TRS-80	Programs and applications ideas for the TRS-80 TRS-80 is a trademark of Tandy Corp
156	$14.90	Small Business Programs	Business Programs in BASIC for use on most microcomputers.

Available direct from us NOW. Dealers please contact us for starter stock. Plans for either large or small dealerships

NEW: MONJANA/1 – THE USER-GUIDED MONITOR ROM FOR COMMODORE CBM

A well-documented, powerful new monitor ROM that any Commodore CBM user can plug into one of the free ROM sockets. At a price of only $98, including an extensive manual, MONJANA/1 offers more user guidance and debugging aids than any other monitor available today. It is indispensable for anyone intending to program his CBM in 6502 machine language. (Trace, link, disassembler, dump, relocate printer option, line assemble and much more.) Order No. 2001

registered trade name of Commodore Business Machines, Inc.

Order No.	Price	Title	Review
2001	$98.00	MONJANA/1 CBM Monitor	Installs in free ROM socket.

NEW: OUR REDYSOFT CASSETTE SOFTWARE

Order No.	Price	Title	Review
3475	$49.00	Assembler for CBM (BASIC and machine language)	Complete assembler including disassembly and link. Cassette and manual
3476	$69.00	Editor/Assembler in machine language.	Fully screen-oriented, scrolling, fail-safe operation.
3999	$34.50	BASIC with I/O for TRS-80	Extended level 1 with I/O and string handling extension. Cassette and manual
8094	$ 1.10	Blank Cassette (Quantity 1)	Highest quality C-10 cassettes.
8095	$ 7.99	Blank Cassettes (Quantity 10)	
8096	$69.00	Blank Cassettes (Quantity 100)	

ELCOMP

3873L Schaefer Avenue
Chino, California 91710
(714) 591-3130

Payment: Check, Visa, Mastercharge
POSTPAID in U.S.A.
California add 6% tax

PUBLISHING, INC.

ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP
ELCOMP ELCOMP